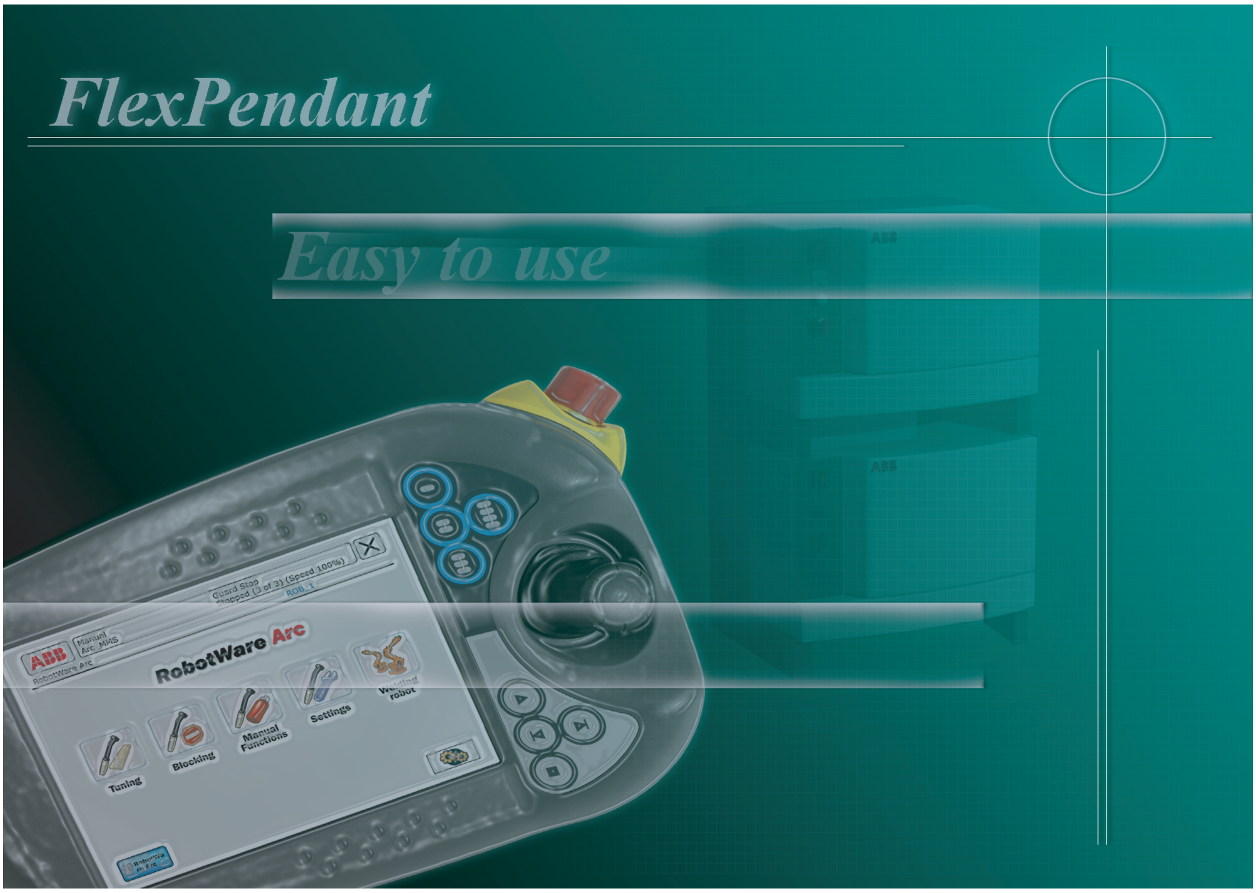
The heart of Robotics

Operator's manual

FlexPendant

IRC5
M2004



Operator’s manual

IRC5 with FlexPendant

M2004

Document ID: 3HAC 16590-1

Revision: B

ABB Automation Technologies AB
Robotics
SE-721 68 Västerås
Sweden

Overview

About this manual

This manual contains instructions for daily operation of IRC5 based robot systems using a FlexPendant.

Usage

This manual should be used during operation.

Who should read this manual?

This manual is intended for:

- operators
- product technicans
- service technicans
- robot programmers

How to read the operator's manual

The operator's manual is structured in the following chapters.

Chapter	Title	Content
1	Safety	Safety instructions and warnings.
2	Welcome to FlexPendant	Descriptions of the FlexPendant and the FlexController.
3	Get started	Descriptions of connections and step-by-step instructions to the most common tasks
4	Navigating and handling the FlexPendant	Descriptions of the FlexPendant's user interface and basic procedures.
5	Jogging	Procedures for jogging.
6	Programming and testing	Procedures for programming and testing, including descriptions of some concepts for programming.
7	Running in production	Procedures for running in production.
8	Handling inputs and outputs, I/O	Procedures for handling I/O.
9	Handling the event log	Procedures for the event log.
10	Systems	Procedures for restart, backup, restore, and configuring systems.
11	Calibrating	Procedures for calibrating the robot system.
12	Changing FlexPendant settings	Procedures for changing the settings for the FlexPendant.
13	Descriptions of terms and concepts	Descriptions of terms and concepts used in robotics.

Continues on next page

Prerequisites

The reader should:

- be familiar with the concepts described in *Getting started - IRC5 and RobotStudio Online*.
- be trained in robot operation.

References

Reference	Document ID
Product manual, procedures - IRC5	3HAC 021313-001
Product manual, references - IRC5	3HAC 021313-001
Getting started - IRC5 and RobotStudio Online	3HAC 021564-001
Operator’s manual - RobotStudio Online	3HAC 18236-1
Trouble shooting manual - IRC5	3HAC 020738-001
Technical reference manual - System parameters	3HAC 17076-1
RAPID reference manual - RAPID overview	3HAC 16580-1
RAPID reference manual - Instructions	3HAC 16581-1
RAPID reference manual - Functions and data types	3HAC 16581-1
RAPID reference manaul - RAPID kernel	3HAC 16585-1
Application manual - Additional axes	3HAC 021395-001
Application manual - Engineering tools	3HAC 020434-001
Application manual - Motion coordination and supervision	3HAC 18154-1
Application manual - Motion functions and events	3HAC 18152-1
Application manual - MultiMove	3HAC 021272-001
Calibration pendulum instruction	3HAC 16578-1
Instructions for levelmeter calibration	3HAC 022907-001

Revisions

Revision	Description
-	First issued. IRC5 M2004. Released with RobotWare 5.04.
A	Second edition. Released with RobotWare 5.05.
B	Third edition. Released with RobotWare 5.06. Organization of chapters restructured to task orientation.

Product documentation, M2004

General

The robot documentation may be divided into a number of categories. This listing is based on the type of information contained within the documents, regardless of whether the products are standard or optional. This means that any given delivery of robot products *will not contain all* documents listed, only the ones pertaining to the equipment delivered.

However, all documents listed may be ordered from ABB. The documents listed are valid for M2004 robot systems.

Hardware manuals

All hardware, robots and controller cabinets, will be delivered with a **Product manual** which is divided into two parts:

Product manual, procedures

- Safety information
- Installation and commissioning (descriptions of mechanical installation, electrical connections and loading system software)
- Maintenance (descriptions of all required preventive maintenance procedures including intervals)
- Repair (descriptions of all recommended repair procedures including spare parts)
- Additional procedures, if any (calibration, decommissioning)

Product manual, reference information

- Reference information (article numbers for documentation referred to in Product manual, procedures, lists of tools, safety standards)
- Part list
- Foldouts or exploded views
- Circuit diagrams

RobotWare manuals

The following manuals describe the robot software in general and contain relevant reference information:

- **RAPID Overview**: An overview of the RAPID programming language.
- **RAPID reference manual part 1**: Description of all RAPID instructions.
- **RAPID reference manual part 2**: Description of all RAPID functions and data types.
- **Technical reference manual - System parameters**: Description of system parameters and configuration workflows.

Continues on next page

Application manuals

Specific applications (e.g. software or hardware options) are described in **Application manuals**. An application manual can describe one or several applications.

An application manual generally contains information about:

- The purpose of the application (what it does and when it is useful)
- What is included (e.g. cables, I/O boards, RAPID instructions, system parameters)
- How to use the application
- Examples of how to use the application

Operator’s manuals

This group of manuals is aimed at those having first hand operational contact with the robot, i.e. production cell operators, programmers and trouble shooters. The group of manuals include:

- **Getting started - IRC5 and RobotStudio Online**
- **Operator’s manual - IRC5 with FlexPendant**
- **Operator’s manual - RobotStudio Online**
- **Trouble shooting manual** for the controller and robot

1 Safety

1.1. About the Safety chapter

Introduction to safety

This chapter describes safety principles and procedures to be used when a robot or robot system is operated.

It does not cover how to design for safety nor how to install safety related equipment. These topics are covered in the Product Manuals supplied with the robot system.

1.2. Applicable safety standards for IRC5

Health and safety standards

The robot complies fully with the health and safety standards specified in the EEC's Machinery Directives.

The ABB robots controlled by the IRC5 conforms to the following standards:

Standard	Description
EN ISO 12100-1	Safety of machinery, terminology
EN ISO 12100-2	Safety of machinery, technical specifications
EN 954-1	Safety of machinery, safety related parts of control systems
EN 775	Manipulating industrial robots, safety
EN 60204	Electrical equipment of industrial machines
EN 61000-6-4 (option)	EMC, generic emission
EN 61000-6-2	EMC, generic immunity

Standard	Description
IEC 204-1	Electrical equipment of industrial machines
IEC 529	Degrees of protection provided by enclosures

Standard	Description
ISO 10218	Manipulating industrial robots, safety
ISO 9787	Manipulating industrial robots, coordinate systems and motions

Standard	Description
ANSI/RIA 15.06/1999	Safety requirements for industrial robots and robot systems
ANSI/UL 1740-1998 (option)	Safety standard for robots and robot equipment
CAN/CSA Z 434-03 (option)	Industrial robots and robot systems - General safety requirements

1.3 Safety terminology

1.3.1. Safety signals, general

General

This section specifies all dangers that may arise from performing the work detailed in the manual. Each danger is detailed in its own section consisting of:

- A caption specifying the danger level (DANGER, WARNING or CAUTION) and the type of danger.
- A brief description of what will happen if the operator/service personnel **do not** eliminate the danger.
- An instruction of how to eliminate the danger to facilitate performing the activity at hand.

Danger levels

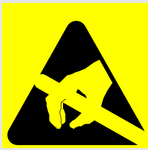
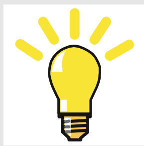
The table below defines the captions specifying the danger levels used throughout this manual.

Symbol	Designation	Signification
 danger	DANGER	Warns that an accident *will* occur if the instructions are not followed, resulting in a serious or fatal injury and/or severe damage to the product. It applies to warnings that apply to danger with, for example, contact with high voltage electrical units, explosion or fire risk, risk of poisonous gases, risk of crushing, impact, fall from height etc.
 warning	WARNING	Warns that an accident *may* occur if the instructions are not followed, that can lead to serious injury, possibly fatal, and/or great damage to the product. It applies to warnings that apply to danger with, for example, contact with high voltage electrical units, explosion or fire risk, risk of poisonous gases, risk of crushing, impact, fall from height etc.
 Electrical shock	ELECTRICAL SHOCK	The electrocution or electrical shock symbol indicates electrical hazards which could result in severe personal injury or death.
 caution	CAUTION	Warns that an accident may occur if the instructions are not followed, that can result in injury and/or damage to the product. It also applies to warnings of risks that include burns, eye injury, skin injury, hearing damage, crushing or slipping, tripping, impact, fall from height etc. Furthermore, it applies to warnings that include function requirements when fitting and removing equipment, where there is a risk of damaging the product or causing a breakdown.

Continues on next page

1.3.1. Safety signals, general

Continued

Symbol	Designation	Signification
 Electrostatic discharge (ESD)	ELECTROSTATIC DISCHARGE (ESD)	The electrostatic discharge (ESD) symbol indicates electrostatic hazards which could result in severe damage to the product.
 Note	NOTE	Note symbols alert you to important facts and conditions.
 Tip	TIP	Tip symbols direct you to specific instructions, where to find additional information or how to perform a certain operation in an easier way.

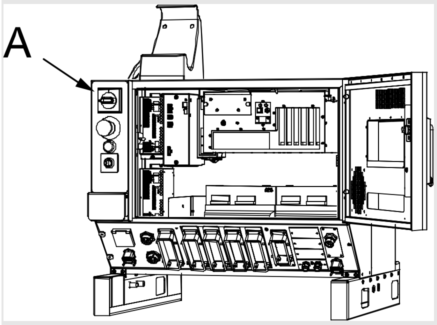
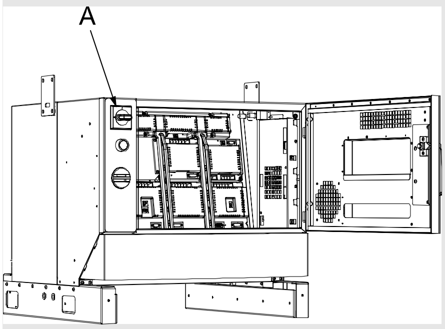
1.3.2. DANGER

1.3.2.1. DANGER - Make sure that the main power has been switched off!

Description

Working with high voltage is potentially lethal. Persons subjected to high voltage may suffer cardiac arrest, burn injuries or other severe injuries. To avoid these dangers, do not proceed working before eliminating the danger as detailed below.

Elimination

Step	Action	Info/Illustration
1.	Switch off the main switch on the Control Module.	 xx0400000978 • A: main switch, control module
2.	Switch off the main switch on the Drive Module.	 en0400001017 • A: main switch, drive module

1.3.2.2. DANGER - Moving manipulators are potentially lethal!

Description

Any moving manipulator is a potentially lethal machine.

When running the manipulator, it may perform unexpected and sometimes irrational movements. However, all movements are performed with great force and may seriously injure any personnel and/or damage any piece of equipment located within the manipulator working range.

Elimination

Step	Action	Info/Illustration
1.	Before attempting to run the manipulator, make sure all *emergency stop equipment* is correctly installed and connected.	Emergency stop equipment such as gates, tread mats, light curtains, etc.
2.	If possible, use the hold-to-run button whenever possible. The hold-to-run button is used in manual mode, not in automatic mode.	How to use the hold-to-run function is detailed in section *How to use the hold-to-run function on page 196*.
3.	Make sure no personnel are present within the manipulator working range before pressing the start button.	

1.3.2.3. DANGER - Manipulator without axes' holding brakes are potentially lethal!

Description

Since the manipulator arm system is quite heavy, especially on larger manipulator models, it is dangerous if the holding brakes are disconnected, faulty, worn or in any way rendered non-operational.

For instance, a collapsing IRB 7600 arm system may kill or seriously injure a person standing beneath it.

Elimination

Step	Action	Info/illustration
1.	If you suspect that the holding brakes are non-operational, secure the manipulator arm system by some other means before working on it.	Weight specifications etc. may be found in the Product Manual of each manipulator model.
2.	If you intentionally render the holding brakes non-operational by connecting an external voltage supply, the utmost care must be taken! **NEVER** stand inside the manipulator working area when disabling the holding brakes unless the arm system is supported by some other means!	How to correctly connect an external voltage supply is detailed in the Product Manual of each manipulator model.

1.3.3. WARNING

1.3.3.1. WARNING - The unit is sensitive to ESD!

Description

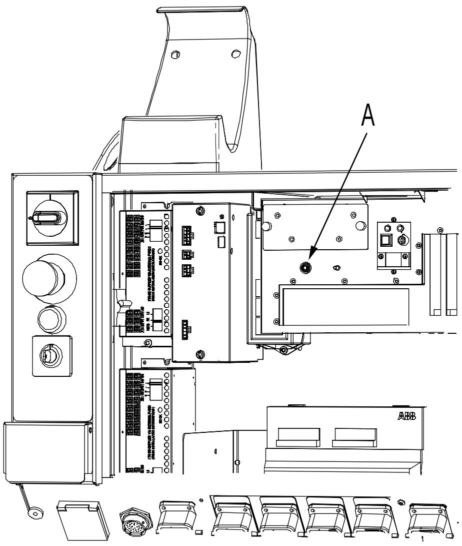
ESD (electro static discharge) is the transfer of electrical static charge between two bodies at different potentials, either through direct contact or through an induced electrical field. When handling parts or their containers, personnel not grounded may potentially transfer high static charges. This discharge may destroy sensitive electronics.

Elimination

Step	Action	Note/Illustration
1.	Use a wrist strap	Wrist straps must be tested frequently to ensure that they are not damaged and are operating correctly.
2.	Use an ESD protective floor mat.	The mat must be grounded through a current-limiting resistor.
3.	Use a dissipative table mat.	The mat should provide a controlled discharge of static voltages and must be grounded.

Location of wrist strap button

The wrist strap button is located on the computer unit in the control module as shown in the illustration below.



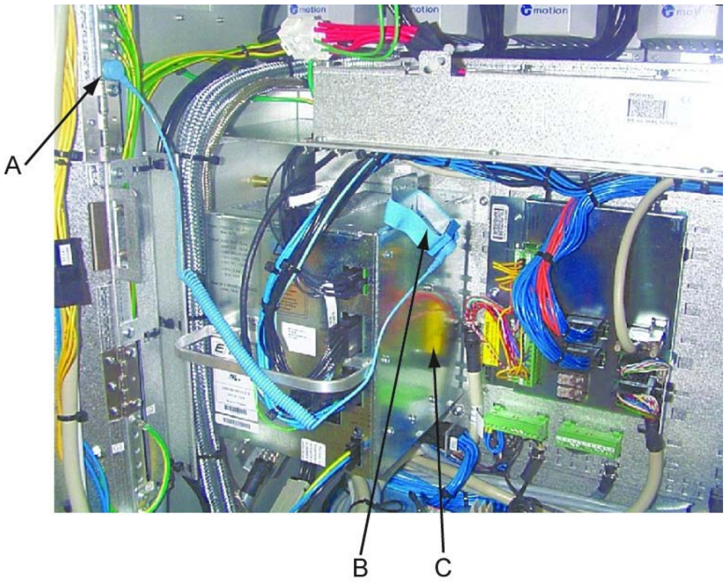
xx0400001061

A	wrist strap button

Continues on next page

Assemble the wrist strap

The picture illustrates how the ESD wrist strap is assembled in the controller.



xx0400001055

A	The strap is fastened to a button on the side of the control module.
B	When not used, the wrist strap is placed on the power supply unit.
C	Power supply unit

1.3.4. What is an emergency stop?

Definition of emergency stop

An emergency stop is a state that overrides any other robot control, disconnects drive power from the robot's motors, stops all moving parts and disconnects power from any potentially dangerous functions controlled by the robot system.

An emergency stop state means that all power is disconnected from the robot except for the manual brake release circuits. You must perform a recovery procedure in order to return to normal operation.

The robot system can be configured so that the state results in either:

- an uncontrolled stop, immediately stopping the robot's action by disconnecting power from its motors
- a controlled stop, stopping the robot's action with power available to its motors so that the robot path can be maintained. When completed, power is disconnected.

Controlled stops are preferred since it minimizes the actions needed to return the robot system back to production. Please consult your plant or cell documentation to see how your robot system is configured.

Classification of stops

The safety standards that regulates automation and robot equipment defines categories in which each type of stop applies:

If the stop is...	..then it is classified as...
uncontrolled	category 0 (zero)
controlled	category 1

Emergency stop devices

In a robot system there are several emergency stop devices that can be operated in order to achieve an emergency stop. There are emergency stop buttons available on the FlexPendant and on the control module. There can also be other types of emergency stops on your robot, consult your plant or cell documentation to see how your robot system is configured.

1.3.5. What is a safety stop?

Definition of safety stops

An emergency stop is a state that overrides any other robot control, disconnects drive power from the robot's motors, stops all moving parts and disconnects power from any potentially dangerous functions controlled by the robot system.

A safety stop means that only the power to the robot's motors is disconnected. There is no recovery procedure. You need only to restore motor power to recover from a safety stop.

The robot system can be configured so that the state results in either:

- an uncontrolled stop, immediately stopping the robot's action by disconnecting power from its motors
- a controlled stop, stopping the robot's action with power available to its motors so that the robot path can be maintained. When completed, power is disconnected.

Controlled stops are preferred since it minimizes the actions needed to return the robot system back to production. Please consult your plant or cell documentation to see how your robot system is configured.

Classification of stops

The safety standards that regulates automation and robot equipment defines categories in which each type of stop applies:

If the stop is...	..then it is classified as...
uncontrolled	category 0 (zero)
controlled	category 1

1.3.6. What is safeguarding?

Definition

Safeguarding are safety measures consisting of the use of safeguards to protect persons from hazards which cannot reasonably be removed or sufficiently eliminated by design.

A safeguard prevents hazardous situations by stopping the robot in a controlled manner when a certain safeguarding mechanism such as a light curtain is activated.

Safeguarded space

The safeguarded space is the space guarded by the guards. For example, a robot cell is safeguarded by the cell door and its interlocking device.

Interlocking devices

Each present guard has an interlocking device which, when activated stops the robot. The robot cell door has an interlock that stops the robot when the door is opened. The only way to resume operation is to close the door.

Safeguarding mechanisms

A safeguarding mechanism consists of a number of guards connected in series. When a guard is activated, the chain is broken and the machine operation is stopped regardless of the state of the guards in the rest of the chain.

1.3.7. Enabling device and hold-to-run buttons

Enabling device

The enabling device is a manually operated constant pressure push-button which, when continuously activated in one position only, allows potentially hazardous functions but does not initiate them. In any other position, hazardous functions are stopped safely.

The enabling device is of a specific type where you must press the push-button only half-way to activate it. In the fully in and fully out positions, robot operation is impossible.

Hold-to-run button

The hold-to-run button allows movement when actuated manually and immediately stops any movement when released. The hold-to-run button can only be used in manual mode.

How to operate the hold-to-run buttons is detailed in section *How to use the hold-to-run function on page 196*.

1.4 How to deal with an emergency

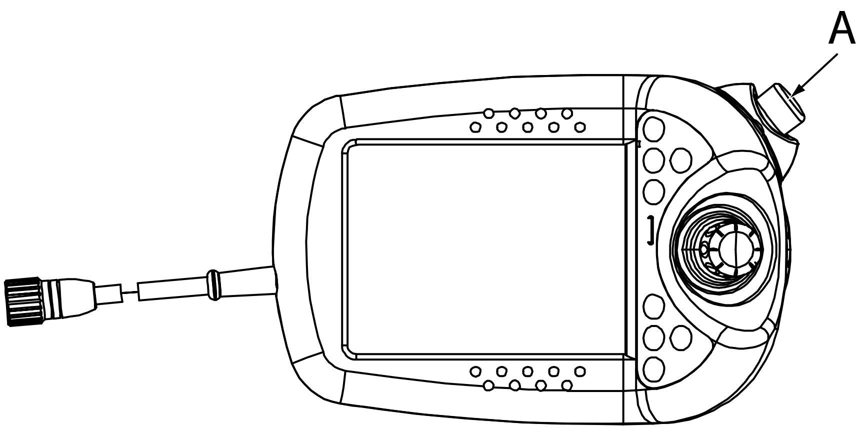
1.4.1. Stop the system

Overview

Stop the system immediately if:

- there are any personnel in the robot working area, while the robot is working
- the robot causes harm to personnel or mechanical equipment

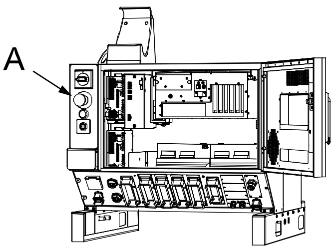
The FlexPendant emergency stop button



xx0300000449

A	Emergency stop button

The controller emergency stop button



xx0300000450

A	Emergency stop button

Other emergency stop devices

The plant designer may have placed additional emergency stop devices in convenient places. Please consult your plant or cell documentation to find out where these are placed.

1.4.2. Release the robot holding brakes

Overview

The robot’s brakes may be manually released as long as power is available. As long as the controller's power switch is in its on position, power is available and applied even if the system is in emergency state.

Battery power

In case of a plant or cell power outage the brake system may be powered by a battery. How to connect the battery is different for each robot model. This is detailed in the Product Manual delivered with the robot.

Brake release buttons

Brake release buttons are placed differently depending on robot type, this is detailed in the Product Manual.

Always learn where the buttons are placed on robot models you work with.

Precautions

Before releasing the brakes verify:

- which way will the arm go?
- how will an entangled object be affected?

A minor damage can easily become serious if the consequences are not considered.

DANGER!

Releasing the brakes is a hazardous action that may cause injury and damage property. It must be done with great care and only when absolutely necessary.

Releasing brakes

Step	Action
1.	If necessary, use an overhead crane, fork lift or similar to secure the robots arms.
2.	Make sure the robot is powered.
3.	Once more, make sure that damage to entangled objects is not extended when brakes are released.
4.	Press the appropriate brake release button to release the brake.

1.4.3. Extinguishing fires

Precautions

In case of a fire always make sure both you and your coworkers are safe before performing any fire extinguishing activities. In case of injury always make sure these are treated first.

Select fire extinguisher

Always use carbon dioxide extinguishers when extinguishing fires in electrical equipment such as the robot or the controller. Do not use water or foam.

1.4.4. Recover from emergency stops

Overview

Recovering from an emergency stop is a simple but important procedure. This procedure ensures that the robot system is not returned to production while maintaining a hazardous condition.

Reset the latch of emergency stop buttons

All push-button style emergency stop devices have a latching feature that must be released in order to remove the emergency stop condition of the device.

In many cases this is done by twisting the push-button as marked, but there are also devices where you pull the button to release the latch.

Reset automatic emergency stop devices

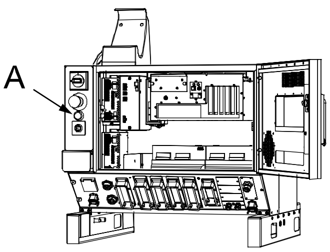
All automatic emergency stop devices also have some kind of latching feature that must be released. Please consult your plant or cell documentation to see how your robot system is configured.

Recover from emergency stops

Step	Action
1.	Make sure the hazardous situation that resulted in the emergency stop condition no longer exists.
2.	Locate and reset the device or devices that gave the emergency stop condition.
3.	Acknowledge the emergency stop event (20202) in the event log.
4.	Press the Motors On button to recover from the emergency stop condition.

The Motors On button

The Motors On button is located on the controller:



en0400000920

A	Motors on button

1.4.5. Return to the programmed path

Overview

Stopping the robot’s movement by removing power to its motors often results in the robot slipping from its programmed path. This may occur after an uncontrolled emergency or safety stop. The allowed slip distance is configured with system parameters. The distance can be different depending on operating mode.

If the robot is not within the configured allowed distance, you may chose to let the robot return to the programmed path or continue to the next programmed point in the path. Then the program execution continues automatically in programmed speed.

1.5 Working in a safe manner

1.5.1. Overview

About the robot

A robot is heavy and extremely powerful regardless of its speed. A pause or longer stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement without warning.

Therefore, it is important that all safety regulations are followed when entering safeguarded space.

About this section

In this section some most basic rules of conduct for you as a robot system user are suggested. However, it is impossible to cover each and every specific situation.

1.5.2. For your own safety

General principles

A few simple principles should be followed in order to operate the robot system safely:

- Always operate the robot system in manual mode if personnel are inside safeguarded space.
- Always bring the FlexPendant along when you enter safeguarded space so that robot control is in your hands.
- Watch out for rotating or moving tools such as milling cutters and saws. Make sure those are stopped before you approach the robot.
- Watch out for hot surfaces both on work pieces as well as on the robot system. The robot’s motors can become fairly hot if run for a long time.
- Watch out for grippers and objects gripped. If the gripper is opened the work piece could fall and cause injuries or damage equipment. The gripper can be very powerful and can also cause injuries if not operated in a safe manner.
- Watch out for hydraulic and pneumatic systems and live electric parts. Even with power off residual energy in such circuits can be very dangerous.

Disconnected FlexPendant

Always put away a disconnected FlexPendant safe from any robot cell or controller to avoid that a disconnected unit is used when trying to stop the robot in a hazardous situation.

CAUTION!

A disconnected FlexPendant should be stored in such a way that it cannot be mistaken for being connected to the controller.

Custom FlexPendant connections

Any means of connecting the FlexPendant except by the supplied cable and its standard connector must not render the emergency stop button inoperative.

Always test the emergency stop button to make sure it works if a custom connection cable is used.

Controller’s access panels

Access panels should only be opened by trained service personnel. There are no parts inside of use to others.

DANGER!

Danger of electrical shock or burn. High voltages inside controller cabinet. The robot and other equipment in the cell are also supplied with high voltages.

1.5.3. Handling of FlexPendant

General instructions

The FlexPendant is a high-quality handheld terminal equipped with highly sensitive state-of-the-art electronics. To avoid malfunctions or damage through improper handling, follow these instructions during operation.

The FlexPendant may only be used for the purposes mentioned in this manual. The FlexPendant was developed, manufactured, tested and documented in accordance with applicable safety standards. If you follow the instructions regarding safety and use as described in this manual, the product will, in the normal case, neither cause personal injury nor damage to machinery and equipment.

CAUTION!

- Turn off the power supply before opening the cable entrance area of the FlexPendant. Otherwise the components could be destroyed or undefined signals could occur.
- Make sure that nobody trips over the cable to prevent the device from falling to the ground.
- Take care not to squeeze and thus damage the cable with any object.
- Do not lay the cable over sharp edges since this can damage the cable sheath.
- When not using the device, hang it on the wall bracket provided for storage.
- Never use sharp objects (e.g. screwdriver) for operating the touch screen. This could damage the touch screen.

CAUTION!

A disconnected FlexPendant should be stored in such a way that it cannot be mistaken for being connected to the controller.

Waste disposal

Observe the national regulations when disposing of electronic components! When replacing components equipped with batteries, please dispose of used batteries properly!

Foreseeable misuse of enabling device

Foreseeable misuse means that it is not allowed to fixate the enabling device in the enabling position. The foreseeable misuse of the enabling device must be restricted.

When releasing and then pressing the enabling device again, make sure to wait for the system to go to Motors Off state before pressing again. Otherwise you will receive an error message.

1.5.4. Safety tools

Safeguarding mechanisms

Your robot system can be equipped with a vast range of safeguards such as door interlocks, safety light curtains, safety mats, and others. The most common is the door interlock of the robot cell that temporarily stops the robot if you open it.

The controller has three separate safeguarding mechanisms, the *general mode safeguarded stop* (GS), the *automatic mode safeguarded stop* (AS) and the *superior safeguarded stop* (SS).

Safeguards connected to...	are...
the GS mechanism	always active regardless of the operating mode.
the AS mechanism	only active when the system is in automatic mode.
the SS mechanism	always active regardless of the operating mode.

Please consult your plant or cell documentation to see how your robot system is configured and where the safeguarding mechanisms are placed and how they work.

Safety supervision

The emergency stop and safeguarding mechanisms are supervised so that any failure is detected by the controller and the robot is stopped until the problem is solved.

Built-in safety safety stop functions

The controller continuously monitor hardware and software functionality. If any problems or errors are detected the robot is stopped until the problem has been solved.

If the failure is...	then...
simple and can easily be solved	a simple program stop is issued (SYSSTOP).
minor and can be solved	a SYSHALT is issued which results in a safety stop.
major, for instance concerns broken hardware	a SYSFAIL is issued which results in an emergency stop. The controller must be restarted in order to return to normal operation.

Restricting the robot's working range

The robot's working range can be restricted by means of mechanical stops or software functions, or by a combination of both.

Please consult your plant or cell documentation to see how your robot system is configured.

1.5.5. Safety in manual reduced speed and manual full speed mode

What is the manual mode?

The manual mode is most often used when creating programs and when commissioning a robot system.

There are two manual modes:

- manual reduced speed mode, usually called manual mode
- manual full speed mode (not available in all markets)

In manual mode, you need to press the enabling device to activate the robot's motors.

What is the manual full speed mode?

In manual full speed mode the robot system can run in full speed. This mode is used when testing programs.

Operating speed

In manual reduced speed mode the robot can only be operated (moved) in reduced speed, 250 mm/s or slower. You should always operate in manual speed whenever working inside safeguarded space.

In manual full speed mode the robot moves in programmed speed. The manual full speed mode should only be used while all personnel are outside safeguarded space and only by specifically trained personnel extra aware of the implied risks.

Bypassed safeguard mechanisms

Automatic mode safeguarded stop (AS) mechanisms are all bypassed while operating in manual mode.

The enabling device

In **manual mode** the robot's motors are activated by the enabling device on the FlexPendant. This way the robot can only move as long as the device is pressed.

The enabling device is designed so that you must press its push-button just half-way to activate the robot's motors. Both in its all-out and full-in positions the robot will not move.

The hold-to-run button

The hold-to-run button allows stepping or running a program in manual mode.

Note that jogging does not require the hold-to-run button, regardless of operating mode.

1.5.6. Safety in automatic mode

What is the automatic mode?

In automatic mode the enabling device is disconnected so that the robot can move without human intervention.

Active safeguard mechanisms

Both the general mode safeguarded stop (GS) mechanisms, the automatic mode safeguarded stop (AS) mechanisms and the superior safeguarded stop (SS) are all active while operating in automatic mode.

Coping with process disturbances

Process disturbances may not only affect a specific robot cell but an entire chain of systems even if the problem originates in a specific cell.

Extra care must be taken during such a disturbance since that chain of events may create hazardous operations not seen when operating the single robot cell. All remedial actions must be performed by personnel with good knowledge of the entire production line, not only the malfunctioning robot.

Process disturbance examples

A robot picking components from a conveyer might be taken out of production due to a mechanical malfunction, while the conveyer must remain running in order to continue production in the rest of the production line. This means, of course, that extra care must be taken by the personel preparing the robot in close proximity to the running conveyor.

A welding robot needs maintenance. Taking the welding robot out of production also means that a work bench as well as a material handling robot must be taken out of production to avoid personnel hazards.

2 Welcome to FlexPendant

2.1. About the Welcome to FlexPendant chapter

Overview

The Welcome to FlexPendant chapter contain an overview on the FlexPendant, the IRC5 controller, and RobotStudio Online.

A basic IRC5 robot system normally consists of a controller, the FlexPendant, RobotStudio Online and one or more robots or other mechanical units. There can also be one or more hardware or software options or additions. This manual describes a basic IRC5 system without options.

2.2. What is a FlexPendant?

Description of FlexPendant

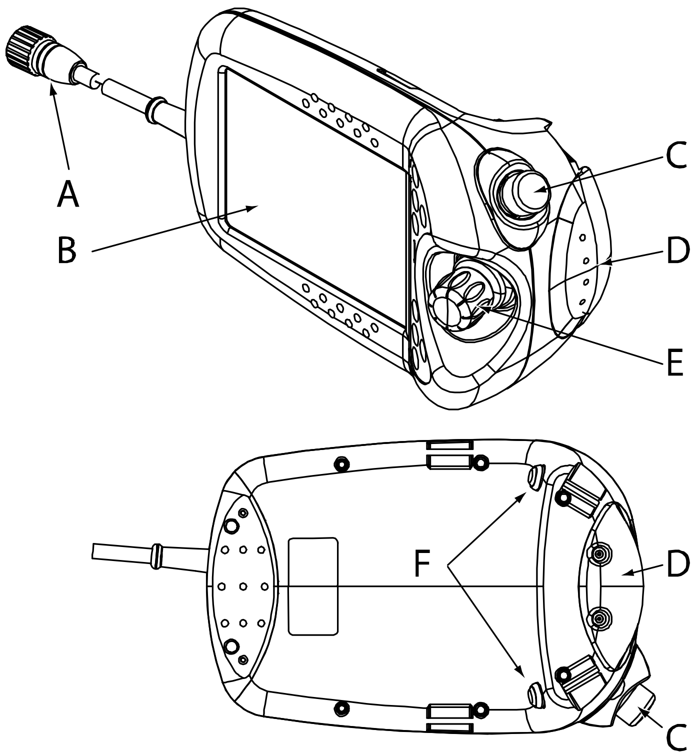
The FlexPendant (occasionally called TPU, or teach pendant unit) is a device for handling many of the functions involved with operating the robot system; running programs, jogging the manipulator, producing and editing application programs, etc.

The FlexPendant consists of both hardware, such as buttons and joystick, and software. The FlexPendant is connected to the controller module through an integrated cable and connector.

NOTE that specific functions may not be performed using the FlexPendant, but only through RobotStudio[Online] . How to perform these are specified in *Operator's manual - RobotStudio Online*.

Main parts

These are the main parts on the FlexPendant.



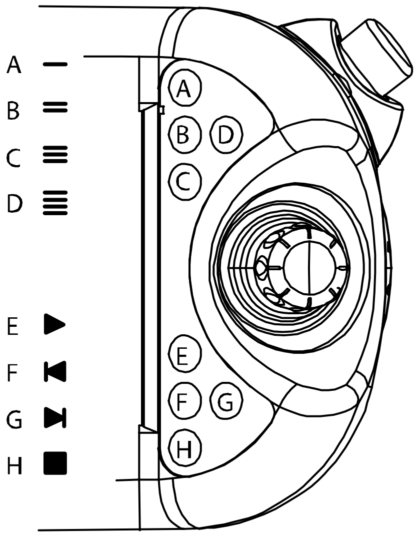
en0300000586

A	Connector
B	Touch screen
C	Emergency stop button
D	Enabling device
E	Joystick
F	Hold-to-run buttons (not included in all systems)

Continues on next page

Hardware buttons

There are a number of dedicated hardware buttons on the FlexPendant. Four of them are programmable and four are pre-programmed.



en0300000587

A	Programmable key 1. How to define its function is detailed in section *Editing programmable keys on page 291* .
B	Programmable key 2. How to define its function is detailed in section *Editing programmable keys on page 291*.
C	Programmable key 3. How to define its function is detailed in section *Editing programmable keys on page 291*.
D	Programmable key 4. How to define its function is detailed in section *Editing programmable keys on page 291*.
E	START button. Starts program execution. In systems without hold-to-run buttons, the Start button is also used for the hold-to-run function.
F	Step BACKWARD button. Steps the program one instruction backwards. In systems *without* hold-to-run buttons, the Backward button is also used for the hold-to-run function.
G	Step FORWARD button. Steps the program one instruction forwards. In systems *without* hold-to-run buttons, the Forward button is also used for the hold-to-run function.
H	STOP button. Stops the program execution.

Continues on next page

Touch screen elements

The illustration shows the touch screen elements of the FlexPendant touch screen.

en0300000588

A	ABB menu
B	Operator window
C	Status bar
D	Close button
E	Task bar
F	Quickset menu

ABB menu

The ABB menu contains programs, configurations, and applications. This is described in section *The ABB menu on page 73*.

Operator window

The Operator window displays messages from the program. This is described in section *Operator window on page 94*.

Status bar

The Status bar displays information about the system and messages. This is described in section *Status bar on page 95*.

Close button

Tapping the close button closes the presently open view or application.

Continues on next page

Task bar

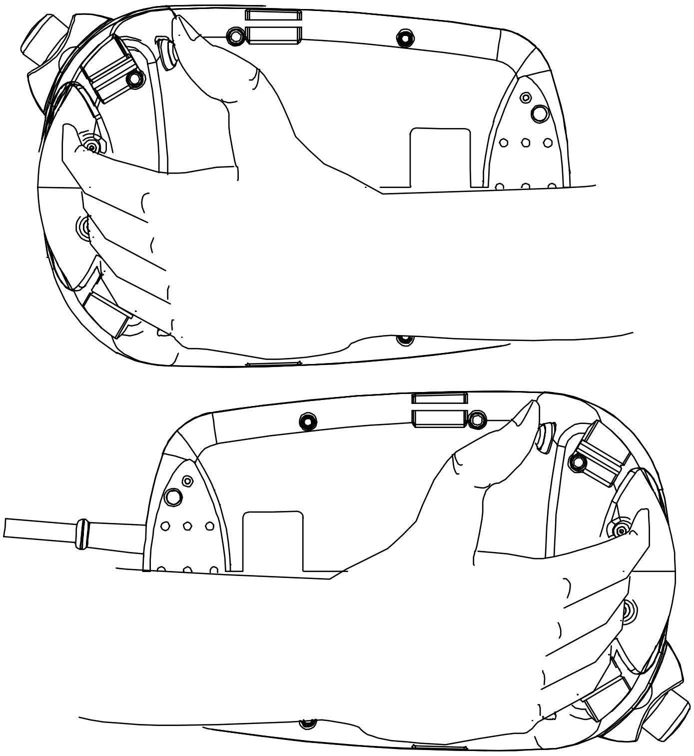
The Task bar displays all open views and applications.

Quickset menu

The Quickset menu contains shortcuts to jogging and settings. This is described in section *The Quickset menu on page 96*.

Left and right handheld

The FlexPendant is set to left handheld on delivery. This can easily be changed to right handheld and back again whenever required.



en0400000913

2.3. What is an IRC5 controller?

The IRC5 controller

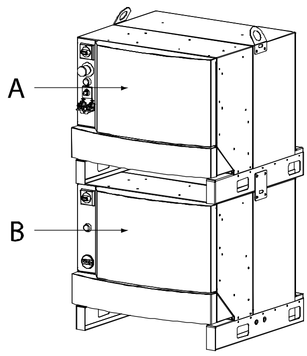
The IRC5 controller contains all functions needed to move and control the robot.

The base variant of the IRC5 controller, M2004, can consist of a single cabinet or be divided into two separate modules; the control module and the drive module. In a single cabinet, the control and drive module are integreted into one single module.

The control module contains all the control electronics such as main computer, I/O boards, and flash memory. The control module runs all software necessary for operating the robot (that is the RobotWare system).

The drive module contains all the power electronics supplying the robot motors. An IRC5 drive module may contain nine drive units and handle six internal axes plus two or additional axes depending on the robot model.

When running more than one robot with one controller (MultiMove option), an extra drive module must be added for each additional robot. However, a single control module is used. Read more about MultiMove in *Application manual - MultiMove*.



xx0400000730

A	Control module
B	Drive module

2.4. What is RobotStudio Online?

Preparations

RobotStudio Online runs on a PC that must be connected either to a controller network or to the controller's service port.

If you are about to install over the controller network you need to know the name or IP address of the controller. You also need to know where the system to be installed is stored, on the PC's hard disk, on a supplied CD or elsewhere.

References

All procedures are detailed in the RobotStudio Online operator's manual.

How to choose what activities to perform using either RobotStudio Online or the FlexPendant is specified in section *When to use the FlexPendant and RobotStudio Online on page 44*.

2.5. When to use the FlexPendant and RobotStudio Online

Overview

For operating and managing the robot, you either use the FlexPendant or RobotStudio^{Online}. The FlexPendant is optimized for handling robot motions and ordinary operation, and RobotStudio^{Online} is optimized for configuration, programming and other tasks not related to the daily operation.

Start, restart and shut down the controller

To...	Use...
Start the controller.	The power switch on the controller's front panel .
Restart the controller.	The **FlexPendant**, **RobotStudio Online** or the power switch on the controller's front panel .
Shut down the controller.	The power switch on the controller's front panel.

Run and control robot programs

To...	Use...
Jog a robot.	The **FlexPendant**
Start or stop a robot program.	The **FlexPendant**

Communicate with the controller

To...	Use...
Acknowledge events.	The **FlexPendant**.
View and save the controller's event logs.	**RobotStudio Online** or the **FlexPendant**.
Back up the controller's software to files on the PC or a server.	**RobotStudio Online** or the **FlexPendant**.
Back up the controller's software to files on the controller	The **FlexPendant**.
Transfer files between the controller and network drives.	**RobotStudio Online** or the **FlexPendant**.

Continues on next page

Program a robot

To...	Use...
Create or edit robot programs in a flexible way. This is suitable for complex programs with a lot of logic, I/O signals or action instructions.	**RobotStudio Online** to create the program's structure and most of the source code and the **FlexPendant** to store robot positions and make final adjustments to the program. When programming, RobotStudio^{Online} provides the following advantages: • A text editor optimized for RAPID code, with auto-text and tool-tip information about instructions and parameters. • Program check with program error marking. • Close access to configuration and I/O editing.
Create or edit a robot program in a supportive way. This is suitable for programs that mostly consist of move instructions.	The **FlexPendant**. When programming, the FlexPendant provides the following advantages: • Instruction pick lists • Program check and debug while writing • Possibility to create robot positions while programming
Add or edit robot positions.	The **FlexPendant**.
Modify robot positions.	The **FlexPendant**.

Configure the robot's system parameters

To...	Use...
Edit the system parameters of the running system.	**RobotStudio Online** or the **FlexPendant**
Save the robot's system parameters as configuration files.	**RobotStudio Online** or the **FlexPendant**
Load system parameters from configuration files to the running system.	**RobotStudio Online** or the **FlexPendant**
Load calibration data.	**RobotStudio Online** or the **FlexPendant**

Create, modify and install systems

To...	Use...
Create or modify a system.	**RobotStudio Online** together with **RobotWare** and a valid **RobotWare Key**.
Install a system on a controller.	**RobotStudio Online**
Install a system on a controller from a USB memory.	The **FlexPendant**.

Continues on next page

Calibration

To...	Use...
Calibrate base frame etc.	The **FlexPendant**
Calibrate tools, work objects etc.	The **FlexPendant**

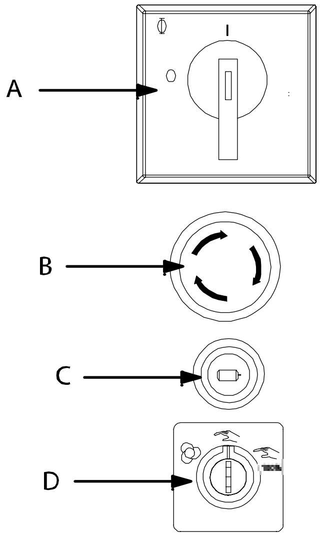
Related information

The table below specifies which manuals to read, when performing the various tasks referred to:

Recommended use...	for details, see manual...	Document number
FlexPendant	Operator's manual - IRC5 with FlexPendant	3HAC 16590-1
RobotStudio Online	Operator's manual - RobotStudio Online	3HAC 18236-1

2.6. Buttons on the controller

Buttons on the control module



en0400000784

Functions for buttons on the control module

This section describes the control module button functions or where to find such information.

Part	Description	Function
A	Main Power ON/OFF	ON/OFF Switch for system shut down. Described in section: • *Start up in automatic mode on page 229* • *Start up in manual mode on page 231*
B	Emergency stop button	
C	Motors On	The different statuses of the motors on lamp is described in the *Trouble shooting manual - IRC5*.
D	Mode switch	Described in section: • *Present operating mode on page 225* • *About the manual mode on page 228* • *About the automatic mode on page 227*

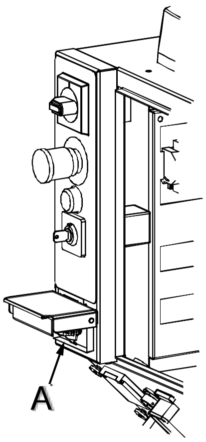
Continues on next page

Ports on the control module

On the front of the control module are also a service port, and optionally a USB port. Both are located below the buttons and may be hidden by a small protective hatch.

The service port is described in section *Connecting a PC to the service port on page 52*.

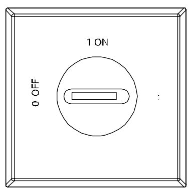
The USB port can be used wih a USB memory device to load or save programs, data, or other information to and from the controller. See section *USB memory information on page 252*.



xx0400001299

A	Service port on control module front

Buttons on the drive module



en0400000797

Functions for buttons on the drive module

This section describes the buttons and their functions on the drive module.

Description	Function
Switch ON/OFF	ON/OFF switch for the drive module alone.

3 Get started

3.1. About the Get started chapter

Overview

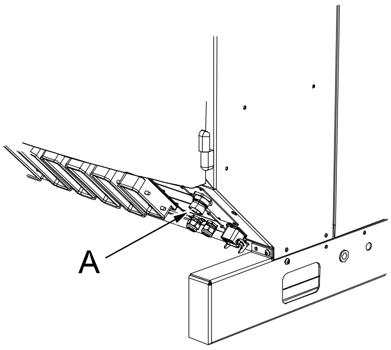
The Get started chapter describes how to connect the FlexPendant to the FlexController, and the network connections. It also describes a number of action scenarios, an overview of ofter performed work tasks with the FlexPendant.

3.2 Connections

3.2.1. Connecting a FlexPendant

Location of FlexPendant connector

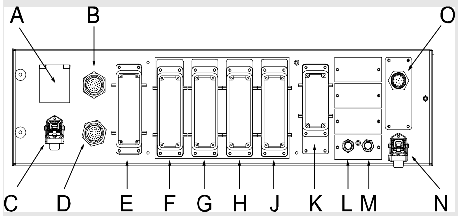
The FlexPendant connector is located as shown below.



xx0400000729

A	FlexPendant socket connector

Connecting a FlexPendant

Step	Action	Illustration
1.	Locate the FlexPendant socket connector on the controller.	 xx0400000931 • O: FlexPendant connector (A22.X1)
2.	Plug in the FlexPendant cable connector.	
3.	Screw the connector lock ring firmly by turning it clockwise.	

3.2.2. Disconnecting a FlexPendant

Disconnecting a FlexPendant

This procedure details how to disconnect a FlexPendant

Step	Action
1.	Finish any ongoing activities that require the FlexPendant to be connected. (For instance path adjustments, calibration, program changes.)
2.	Shut down the system. If the system is not shut down when disconnecting the FlexPendant it will go to the emergency stop state.
3.	Unscrew the connector cable counter clockwise.
4.	Store the FlexPendant safely away from any robot system.

3.2.3. Connecting a PC to the service port

NOTE!

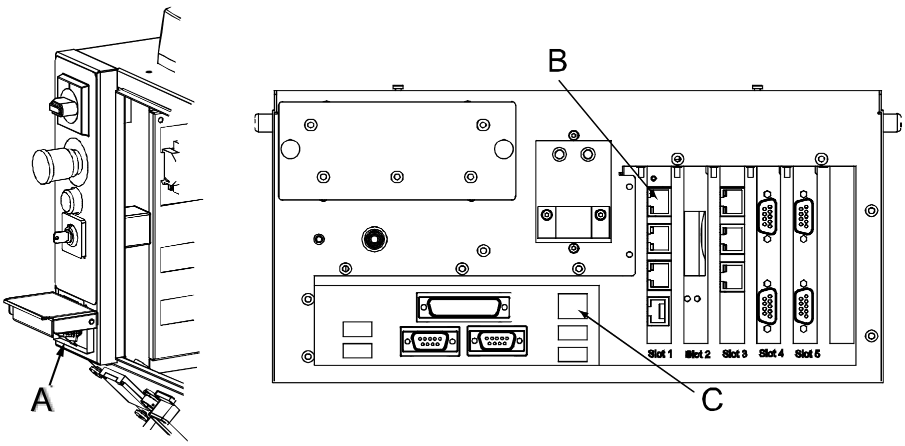
The service port shall only be used for direct connection to a PC as described in this procedure. It must not be connected to a LAN (local area network), since it has a DHCP server that automatically distributes IP addresses to all units connected to the LAN. Contact your network administrator if you need more information.

CAUTION!

When a cable is connected to the service port on the Control Module front and the service hatch is opened, the controller will not comply with the requirements of protection class IP54.

Ports

The illustration below shows the two main ports on the computer unit: the Service Port and the LAN port. Make sure the LAN (factory network) is **not** connected to any of the service ports!



xx0400001299

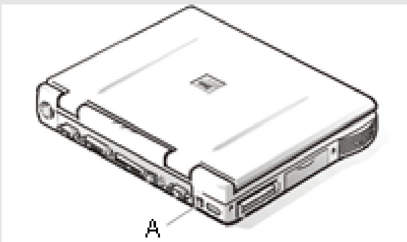
A	Service port on the Control Module front
B	Service port on computer unit (connected to Service port on the Control Module front through a cable)
C	LAN port on computer unit (connects to factory LAN)

Connections to ports may be done as detailed below:

Connection to/from:	Detailed in section:
Connecting the Control Module to the factory LAN	Proceed as detailed in section *Installation* in the *Product Manual, IRC5.*
Connecting a PC to the Control Module service port.	Proceed as detailed below!

Continues on next page

Connecting a pc to the service port

Step	Action	Illustration
1.	Make sure the network setting on the PC to be connected is correct.	Refer to the system documentation for your PC, depending on the operative system you are running. The PC must be set to “Obtain an IP address automatically”.
2.	Use the delivered category 5 Ethernet crossover boot cable with RJ45 connectors.	The cable is delivered in the RobotWare product box.
3.	Connect the network cable to the network port of your PC	 xx0400000844 • A: network port The placement of the network port may vary depending on the pc model.
4.	Connect the boot cable to the service port, placed on the control module front.	

3.2.4. Set up the network connection

When do I need to setup the network connection?

You need to setup the controller's network connection when the controller is connected to a network for the first time or when the network addressing scheme changes.

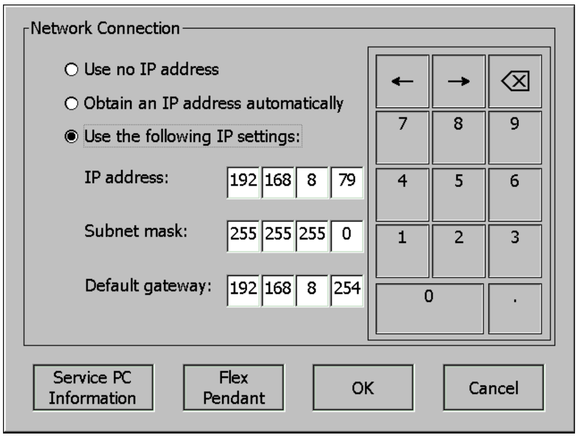
Preparations

If an IP address is to be obtained automatically, make sure there is a server running that supplies the network with IP addresses (a DHCP server). Otherwise you will not be able to access the controller via the controller network.

It is still possible to access the controller via the service PC connection.

Network connection dialog box

The illustration shows the network connection dialog box.



en0400000902

Set up the network connection

Regardless of how you choose to set up the network connections, the first steps are common:

Step	Action	Info
1.	You may reach the boot application by performing an X-start.	How to perform an X-start is detailed in section *Restart and select another system (X-start) on page 258*.
2.	In the boot application, tap **Settings**. The network connection dialog is displayed.	
3.	If you choose to use no IP address, the tap **Use no IP address** . Otherwise, proceed below!	In some cases it can be useful to disconnect the controller from the network, without disconnecting the network cable. Without IP address the controller cannot be accessed from other equipment on the same network.
4.	If you choose to obtain an IP address automatically, the tap **Obtain an IP address automatically**. Otherwise, proceed below!	

Continues on next page

Step	Action	Info
5.	If you choose to use a fixed IP address, tap **Use the following IP address**. Enter a valid IP address, a valid subnet mask, and the IP address of the default gateway to use.	
6.	Tap **OK** to save the new setting.	
7.	In the boot application, tap **Restart Controller** to restart the controller and use the new setting.	

3.3 Action scenarios

3.3.1. About the Action Scenarios chapter

Overview

This chapter contains brief procedures detailing a number of typical actions a typical user may perform. It also contains many references to detailed information about the same subjects.

The brief information given, is intended to be used directly by experienced users, while the references may be more adequate for novices and for training purposes.

More information

Note that there may be more information available than the one referred to in the procedures.

Information about:

- a specific menu is described in chapter *Navigating and handling FlexPendant on page 71*.
- a specific button on the FlexPendant is described in *What is a FlexPendant? on page 38*.
- a specific button is described in chapter *What is an IRC5 controller? on page 42* for tasks performed using the controls on the controller modules.
- how to perform a specific task is detailed in the tasks chapters, e.g. *Programming and testing on page 133* or *Running in production on page 219*.

3.3.2. System start up

Prerequisites before start up

This procedure details the main steps required to start the system when the power has been switched off.

All information is based on the assumption that *working system software has already been installed* on the robot controller, as the case would be at first start-up directly after delivery.

No information is given in this manual about how to connect the controller to a LAN (Local Area Network), but this is detailed in the Installation section of the *Product Manual - IRC5*.

Note that there may be more information available than the one referred to in the procedure.

System start up

This procedure details all required steps to start the system for the first time. For everyday startup, step 4 is normally the only required step.

Step	Action	Info
1.	Install the robot equipment.	Mechanical installation and electrical connections between manipulator and controller cabinet is described in the *Product Manual* of the robot and controller respectively.
2.	Make sure the safety circuits of the system are properly connected to the robot cell or have jumper connections installed (if required).	How to connect the safety circuits is detailed in the robot's *Product Manual*.
3.	Connect the FlexPendant to the controller cabinet.	The FlexPendant and its major parts and functions are detailed in section *What is a FlexPendant? on page 38* How to connect the FlexPendant to the cabinet is detailed in section *Connecting a FlexPendant on page 50*
4.	Switch the power on.	Use the main switch on the control module.
5.	If the controller or manipulator have been replaced with spare parts, make sure the calibration values, revolution counters and serial numbers are updated correctly.	Normally, only the revolution counters require updating, which is to be performed as detailed in section *Updating revolution counters on page 274*. If required, transfer the calibration data from the serial measurement board as detailed in *Serial Measurement Board memory on page 280* for systems *without* the AbsAcc option. If required, enter the calibration data as detailed in *Loading calibration data using the FlexPendant on page 276* for systems *with* the AbsAcc option.
6.	This step is only required if the robot system will be connected to a network. Perform an *X-start*. The Boot Application is started.	Detailed in section *Restart and select another system (X-start) on page 258*.

Continues on next page

Step	Action	Info
7.	This step is only required if the robot system will be connected to a network. Use the boot application to: • set the IP address of the controller cabinet • set the network connections • select the system • restart the system The system is restarted.	How to use the boot application is detailed in section *Using the boot application on page 254*. At this point, a single system is available.
8.	Install RobotStudio^Online on a PC.	Proceed as detailed in section *Installing RobotStudio Online* in the *Operator's manual - RobotStudio Online*. RobotStudio^Online is used to create a system to run on the controller, but at this point (prior to the first start-up) a system is already installed by the manufacturer.
9.	Connect the controller to a PC (through the service port) or to the network (if used). Connect a PC to the Control Moduleservice port.	Proceed as detailed in section *Connecting a PC to the service port on page 52*. Also see section *Set up the network connection on page 54*.
10.	Start RobotStudio^Online on the PC.	Proceed as detailed in section *Accessing a controller from RobotStudio Online* in the *Operator's manual - RobotStudio Online*.
11.	Restart the controller.	
12.	The robot system is now ready for operation.	

3.3.3. Jogging

Jogging

This procedure details the main steps required to jog the robot.

The term Jogging is described in section *Introduction to jogging on page 105*.

Note that there may be more information available than the one referred to in the procedure.

Step	Action	Info
1.	It is possible to jog the robot under the following conditions: • The system has been started as detailed in this manual. • no programmed operation is running • the system is in Manual mode. • the enabling device is pressed and the system is in Motors On state	The Manual mode is described in section *About the manual mode on page 228*. Starting in the Manual mode is detailed in section *Start up in manual mode on page 231*. How to switch to manual mode is detailed in section *Switching from automatic to manual mode on page 235*.
2.	Many of the mechanical units connected to the controller may be jogged.	How to determine which mechanical unit to jog is detailed in section *Selecting mechanical unit on page 108*.
3.	The robot may be jogged in several ways, in different coordinate systems. First, determine in which way you want to jog.	The difference between different types of jogging is detailed in section *Introduction to jogging on page 105*. How to jog the robot axis by axis is detailed in section *Jog axis by axis on page 114*. The robot may be jogged in: • *Jog in base coordinates on page 116* • *Jog in tool coordinates on page 120* • *Jog in world coordinates on page 117* • *Jog in work object coordinates on page 119*
4.	Once a mechanical unit has been selected, its axes may be jogged in different ways. These ways may be selected using the QuickSet menu.	
5.	Define the working range for the robot/ robots as well as for any other pieces of equipment working in the robot cell.	The robot's working range is defined by system parameters. See section *Configuring system parameters on page 270* or *Technical reference manual - System parameters*.

Continues on next page

Step	Action	Info
6.	Jog the manipulator using the joystick on the FlexPendant.	The FlexPendant and its various parts and sections are described in section *What is a FlexPendant? on page 38*. The joystick and how to map the directions of it, is detailed in section *Selecting motion mode on page 110*. How to prevent causing manipulator movements in certain directions while jogging, is detailed in section *Locking the joystick in specific directions on page 122*. There might be restrictions to how you can jog, see section *Restrictions to jogging on page 106*.
7.	In some cases, more than one manipulator may be jogged simultaneously. This requires the MultiMove option to be installed.	How to jog multiple manipulators is detailed in section *Coordinated jogging on page 107*.

3.3.4. Using RAPID programs

Using the RAPID program

This procedure describes the main steps required in creating, saving, editing and debugging any RAPID program.

Note that there is more information available, than the one referred to in the procedure. The concept RAPID program is described in section *The structure of a RAPID application on page 134*.

Step	Action	Info
1.	Start by creating a RAPID program.	How to create a RAPID program is detailed in section *Handling of programs on page 165*.
2.	Edit your program.	Proceed as detailed in section *Handling of instructions on page 176*.
3.	To simplify programming and keep an overview of the program, you may want to divide the program into more than one module.	The module concept is described in section *The structure of a RAPID application on page 134*. How to view, add, or delete a module is detailed in section *Handling of modules on page 168*.
4.	To further simplify programming, you may want to divide the module into more than one routine.	The routine concept is described in section *The structure of a RAPID application on page 134*. How to add or delete a routine is detailed in section *Handling of routines on page 172*.
5.	When programming you may want to work with: • Tools • Work objects • Payloads	Also read the following sections: • *Creating a tool on page 142*. • *Creating a work object on page 154*. • *Creating a payload on page 161*.
6.	Based on the four elements specified above, program execution may automatically be displaced to better suit e.g. the tools as they wear down, etc.	
7.	In order to deal with potential errors that may occur during program execution, you may want to create an error handler.	Error handlers are described in *RAPID overview*.
8.	After completing the actual RAPID program, it will require testing before being put into production.	Proceed as detailed in section *Testing on page 196*.

Continues on next page

Step	Action	Info
9.	After test running your RAPID program, it may require altering. You may want to modify, or tune, programmed positions, the TCP positions, or paths.	How to modify positions while the program is running is described in section *HotEdit on page 73*. How to modify positions in manual mode is described in section *Modifying positions on page 184*.
10.	Programs that are no longer required may be removed.	See *Deleting programs from memory on page 192*. Also see *Deleting programs from hard disk on page 194*.

Running the program

This procedure specifies how to use an existing RAPID program.

Step	Action	Info
1.	Load an existing program.	Described in section *Starting programs on page 219*.
2.	When starting program execution, you may choose between running the program once, or running it continuously.	Described in section *Quickset menu, Run Mode on page 201*.
3.	Once the program has been loaded, you may start program execution.	Described in section *Starting programs on page 219* and in *Using multitasking programs on page 222*.
4.	After program execution is completed, the program may be stopped.	Proceed as detailed in section *Stopping programs on page 221*.

3.3.5. Working with inputs and outputs

Working with inputs and outputs

This procedure details the main steps required to set outputs, read inputs and configure I/O units.

Note that there may be more information available than the one referred to in the procedure.

Step	Action	Info
1.	You may want to create a new I/O.	I/O signals are created using system parameters, see section *Configuring system parameters on page 270*.
2.	Before using any input or output, the system must be configured to enable the I/O functions.	Configuring the system is done when creating the system. How to do this is detailed in *Operator's manual - RobotStudio Online*.
3.	You may set a value to a specific *digital output*.	Proceed as detailed in section *Simulating and changing signal values on page 238*.
4.	You may set a value to a specific *analog output*.	Proceed as detailed in section *Simulating and changing signal values on page 238*.
5.	You may view the status of a specific *digital input*.	Proceed as detailed in section *Simulating and changing signal values on page 238*.
6.	You may view the status of a specific *analog input*.	Proceed as detailed in section *Simulating and changing signal values on page 238*.
7.	Safety signals.	Signal explanation is detailed in *Safety I/O signals on page 240*
8.	How to edit an I/O.	Proceed as detailed in section*Simulating and changing signal values on page 238*.

3.3.6. Backup and restore

Backup and restore

The contents of a typical backup is specified in section *What is saved on backup? on page 265*. How to perform the backup is detailed in section *Backup the system on page 267*.

Re-introducing the previously saved memory contents from the backup into the robot controller is called *performing a restore*. How to perform the restore is detailed in section *Restore the system on page 268*.

Information about starts is described in *Restart overview on page 253*.

Note that there may be more information available than the one referred to above.

3.3.7. Running in production

Running in production

This instruction details the main steps useful when running the system in automatic mode (production mode).

Note that there may be more information available than the one referred to in the procedure.

Step	Action	Info
1.	Start the system as detailed in section *System start up on page 57*.	
2.	If the system is using UAS, User Authorization System, the user must log into the system before starting operation.	How to log in is described in section *How to logout and login on page 103*.
3.	Load a program.	How to load a program is described in *Handling of programs on page 165*.
4.	Before starting system choose mode to start in on the FlexController.	How to choose mode is described in section *Switching from manual to automatic mode on page 234*.
5.	Start by pressing the Start button on the FlexPendant.	The FlexPendant's hardware buttons are described in *What is a FlexPendant? on page 38*.
6.	The controller system communicates with the operator through messages displayed on the FlexPendant screen. Messages can be either event messages or RAPID instructions, e.g. `TPWrite`. Event messages describe an event occurring within the system, and is saved in an event log.	The basic concepts are described in section *Accessing the event log on page 243*. RAPID instructions `TPReadFK` and `TPWrite` are described in *RAPID reference manual - Instructions*.
7.	In manual mode, the Modify Position function allows the operator to make adjustments to the robot positions in a RAPID program. The HotEdit function allows the operator to make adjustments to programmed positions in both automatic and manual mode.	How to modify position is described in sections *Modifying positions on page 184* and in *HotEdit on page 73*
8.	In a production process you may want to stop the robot.	How to stop production is described in section *Stopping programs on page 221*
9.	In the Production Window you can supervise the ongoing process.	The Production window is described in section *Production window on page 79*
10.	When ending operation, the user should log out.	How to log in is described in section *How to logout and login on page 103*

3.3.8. Granting access for RobotStudio Online

About write access on the controller

The controller only accepts one user with write access at a time. Users in RobotStudio Online can request write access to the system and this request is accepted or rejected on the FlexPendant. This can only be done in manual mode.

Granting access for RobotStudio Online

This procedure describes how to grant access for RobotStudio Online.

Step	Action
1.	When a user in RobotStudio Online requests access, a message is displayed on the FlexPendant. Decide whether to grant or reject access. If you want to grant access, then tap **Grant**. The user holds write access until he disconnects or until you reject the access. If you want to deny access, then tap **Reject**.
2.	If you have granted access and want to revoke the access, tap **Reject**.

3.3.9. Upgrading

Upgrading

This procedure details the main steps required to correctly upgrade the system. By upgrading we mean changing hardware, such as replacing circuit board with newer versions, as well as loading software with later releases.

Note that there may be more information available than the one referred to in the procedure.

Type of upgrade	Info
When replacing circuit boards such as buses, I/O boards, etc., with newer versions, the system will automatically reflash the unit.  xx0100000003 During reflashing, the system may restart several times, and it is vital not to shut down the system, or in any other way interrupt the automatic process.	What happens during reflashing is detailed in section *Reflashing firmware and FlexPendant on page 263*.
When upgrading the robot or controller mechanically, fitting instructions are normally delivered with the kit. If no such instruction are provided, useful information may be found in the Repair section of the *Product Manual* of the equipment in question.	
When upgrading the system software, the system must be changed in order to reflect the additions. A new license key may be required.	How to modify an existing system is detailed in section *How to Modify a System* in the *Operator's manual - RobotStudio Online*. How to create a new system is detailed in section *Creating a new system* in the *Operator's manual - RobotStudio Online*.

3.3.10. Installing software options

Installing software options

The main steps required to correctly install a generic software option or option package is described in *Operator's manual - RobotStudio Online*.

3.3.11. Shutting down

Shutting down

This procedure describes how to shut down the system and turn off power.

Step	Action	Info
1.	Stop all running programs.	
2.	Shut down the system using the On/Off switch on the FlexController.	
3.	If you want to protect the FlexPendant you can unplug it and store it elsewhere when the system has shut down.	How to disconnect the FlexPendant from the controller is detailed in section *Disconnecting a FlexPendant on page 51*.

3.3.12. General procedure when trouble shooting

Types of faults

Faults occurring in the robot system may be of two categories:

- Faults detected by the built-in diagnostics system. These faults are described in section *Event log messages* in *Trouble shooting manual - IRC5*.
- Faults NOT detected by the built-in diagnostics system. These faults are described in section *Other types of faults* in *Trouble shooting manual - IRC5*.

Faults causing error message on the FlexPendant

The control system is supplied with diagnostic software to facilitate trouble shooting and to reduce downtime. Any errors detected by the diagnostics are displayed in plain language with a code number on the FlexPendant.

All system and error messages are logged in a common log in which the last 150 messages are saved. The log can be accessed from the Status bar on the FlexPendant.

To facilitate trouble shooting, it is important that some basic principles are followed. These are specified in *Trouble shooting principles* in *Trouble shooting manual - IRC5*.

Step	Action	Info
1.	Read the error message displayed on the FlexPendant and follow any instructions given.	How to interpret the messages is detailed in section *Overview, event log messages* in *Trouble shooting manual - IRC5*.
2.	Was the information given on the FlexPendant enough to solve the problem? If yes; resume operation. If no; proceed below.	
3.	If relevant, check the LEDs on the units.	Each unit is thoroughly described in section *Unit LEDs* in *Trouble shooting manual - IRC5*, including a description of its LEDs.
4.	If relevant, check the cables, etc., with help of the circuit diagram.	All relevant document numbers are specified in section *Document references, IRC5* in *Trouble shooting manual - IRC5*.
5.	Replace, adjust or fix as detailed in the Repairs instruction if required.	All relevant document numbers are specified in section *Document references, IRC5* in the *Trouble shooting manual - IRC5*.

Faults NOT causing error messages on the FlexPendant

These faults are not detected by the diagnostic system and are handled in other ways. The way the symptom of the fault is observed greatly influences the type of fault. Instructions are given in section *Other types of faults* in *Trouble shooting manual - IRC5*.

To trouble shoot faults NOT causing error messages on the FlexPendant, follow steps 3 and 4 in the procedure in *Faults causing error message on the FlexPendant on page 70*.

4 Navigating and handling FlexPendant

4.1. About the Navigate and handle FlexPendant chapter

Overview

This chapter describes the touch screen elements on the FlexPendant, i.e. the menues.

Each menu is described in overview, with references to more information on how to use the functions. Note that this manual only covers menues in the basic RobotWare system, without options. All options are described in the application manuals.

4.2. Overview, personalizing the FlexPendant

Personalizing

The FlexPendant can be personalized in a number of ways. How to do this is described in the following sections:

How to:	is described in section:
change the language used in windows and dialogs	*Changing language on page 290*.
change the display's brightness and contrast	*Changing brightness and contrast on page 285*.
rotate the FlexPendant for Left/Right handheld use	*Switching between left and right handheld FlexPendant on page 286*.
configure views for program start and user authorization system	*Configuring view settings on page 295*.
recalibrate the touch screen	*Calibrating the touch screen on page 298*.
configure programmable keys	*Editing programmable keys on page 291*.
configure most common I/O list	*Configuring Most Common I/O on page 289*.
change background image	*Changing background image on page 297*.
change the date and time	*Changing date and time on page 288*.

4.3 The ABB menu

4.3.1. HotEdit

HotEdit

HotEdit is a function for editing programmed positions. This can be done in all operating modes, even while the program is running.

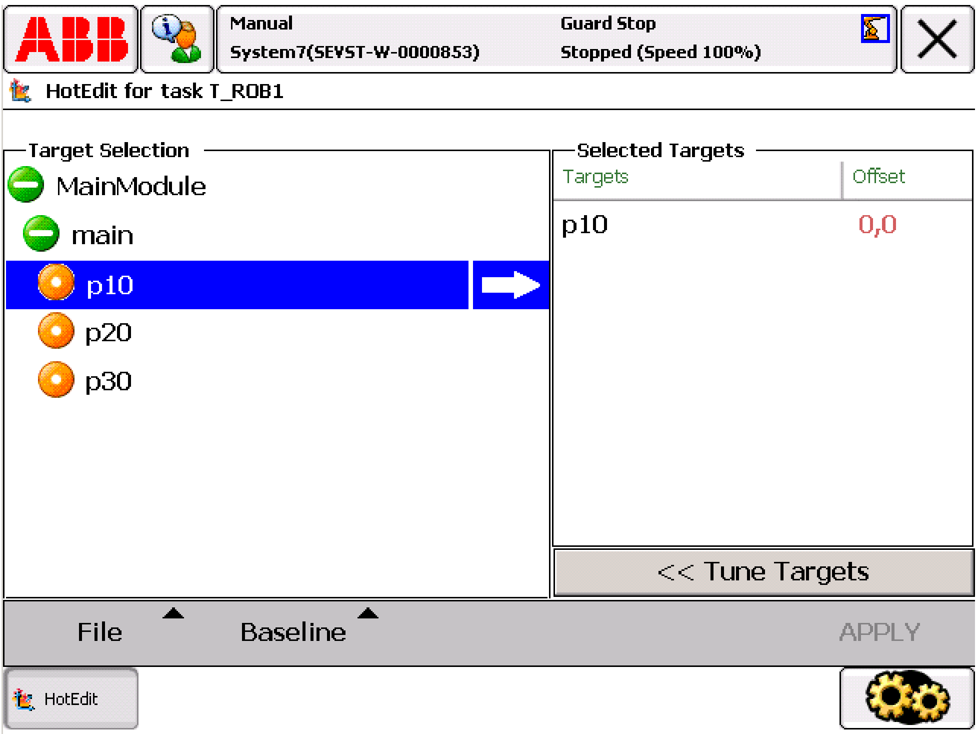
When modifying a position, both the coordinates and the orientation can be changed. Note that the changed offset values are not stored in the original baseline of the program until a Commit command has been applied. The changed values are however used directly in the loaded program.

HotEdit can only be used for positions of the type robtarget.

If a position is changed during operation, the current path may be affected. It is not recommended to modify positions close to the program pointer or the motion pointer since it is hard to predict when changes will take effect if they occur between the two pointers. Therefore it is important to be certain of where in the program the robot is before changing any values while the program is running.

The functions available in HotEdit may be restricted by the user authorization system, UAS.

Illustration of HotEdit menu



en0500001542

Continues on next page

Functions available in HotEdit

Target selection	Lists all named positions in a tree view. Select positions and add them to the section by tapping the arrow. Note that if a position is used in more than one routine, it will appear in all places used and any changes made to the offset will be the same for everywhere it is used.
Selected targets	Lists all selected positions and their current offset. Tap the trash can to the right of the position name to remove them from the selection.
File	You can save and load selections of often used positions using the File menu. If your system uses UAS, this may be the only way to select positions for editing.
Baseline	The baseline menu is used to apply or reject changes to the original (baseline) system, for both HotEdit and position modifications in the Program editor. For position modifications in the Program editor, the baseline menu is can only be used for systems with absolute limit modpos, where the allowed change distance is limited. See system parameters for the type *ModPos Settings* in *Technical reference manual - System parameters*. The original values are not changed until a Commit command has been applied. To apply or reject the changes made to offset values, tap: • **Restore to original** to discard all changes to the currently selected target positions • **Restore entire program to original** to discard all changes to target positions (also applies to changes made in the program editor) • **Commit to current** to apply all current changes to the selected target positions • **Commit entire program to current** to apply all changes to target positions (also applies to changes made in the program editor)
Tune targets	Tap **Tune targets** to display a keyboard for editing the offset values (coordinates and orientation). The offset value is the length of the vector calculated from the original value and the changed value (x, y, and z, and orientation) in the Tune targets menu.
APPLY	Tap **APPLY** to apply changes made in the Tune targets menu. Note that this does not change the original values for the positions!

Related information

Positions can also be modified by jogging the robot to the new position, see section *Modifying positions on page 184*.

RAPID reference manual - Functions and datatypes.

Technical reference manual - System parameters, chapter *Controller*.

What is “the memory”? on page 248.

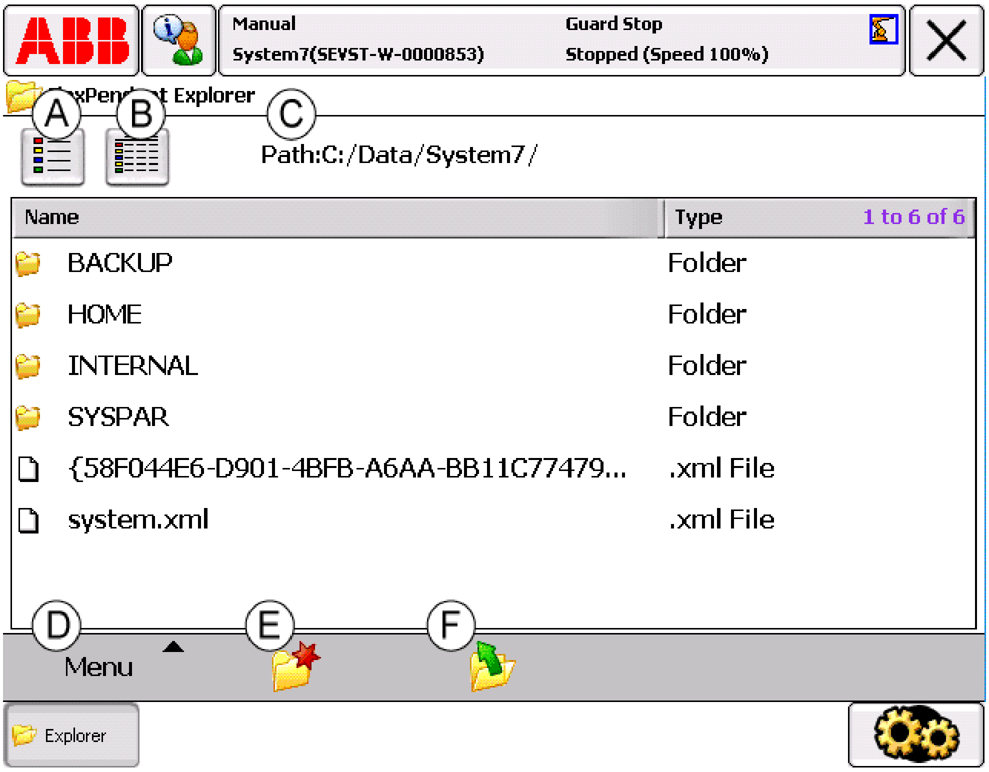
4.3.2. FlexPendant Explorer

About FlexPendant Explorer

The FlexPendant Explorer is a file manager, similar to Windows Explorer, with which you can view the file system on the controller. You can also rename, delete, or move files or folders.

Illustration FlexPendant Explorer

The illustration details the FlexPendant Explorer.



en0400001130

Parts FlexPendant Explorer

The table below describes the parts in the illustration above.

Part	Description	Function
A	Simple view	Tap to hide type in the file window
B	Detailed view	Tap to show type in the file window
C	Path	Displays the directory path
D	Menu	Tap to display menu with functions for file handling
E	New folder	Tap to create new folder in present folder.
F	Up one level	Tap to change to parent folder

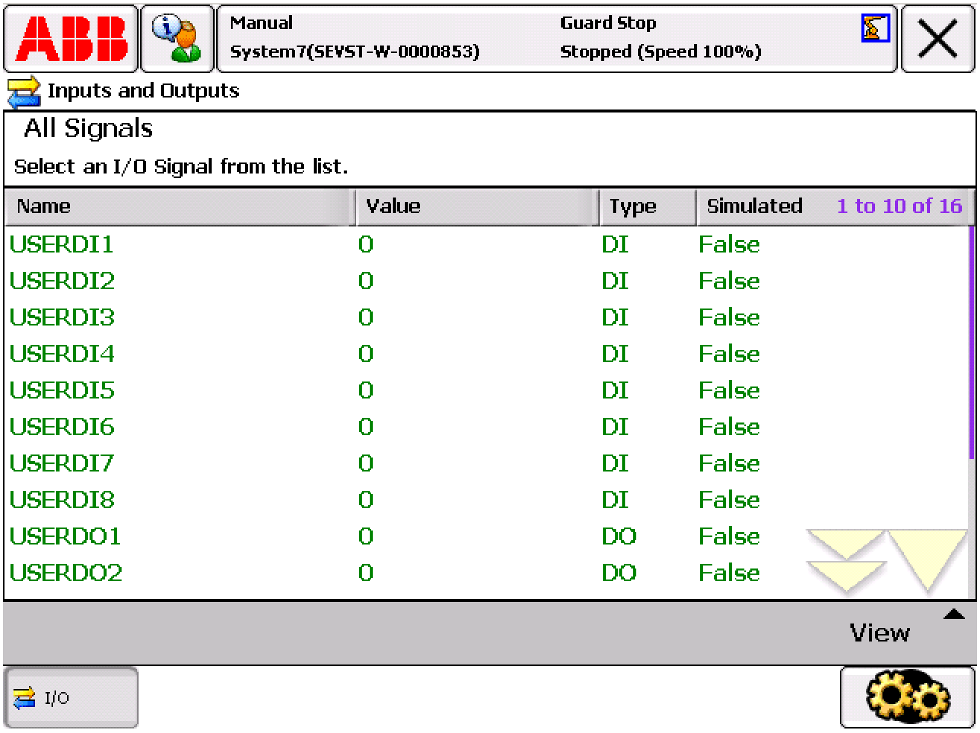
4.3.3. Inputs and outputs, I/O

Inputs and outputs

Inputs and outputs, I/O, are signals used in the robot system. Signals are configured with system parameters, see section *Configuring system parameters on page 270*.

The Inputs and outputs menu

This illustration details the Inputs and outputs menu.



en04000000770

What is a signal

An I/O signal is the logical software representation of an I/O signal located on a fieldbus unit that is connected to a fieldbus within the controller.

By specifying a signal, a logical representation of the real I/O signal is created. The signal configuration defines the specific system parameters for the signal that will control the behavior of the signal.

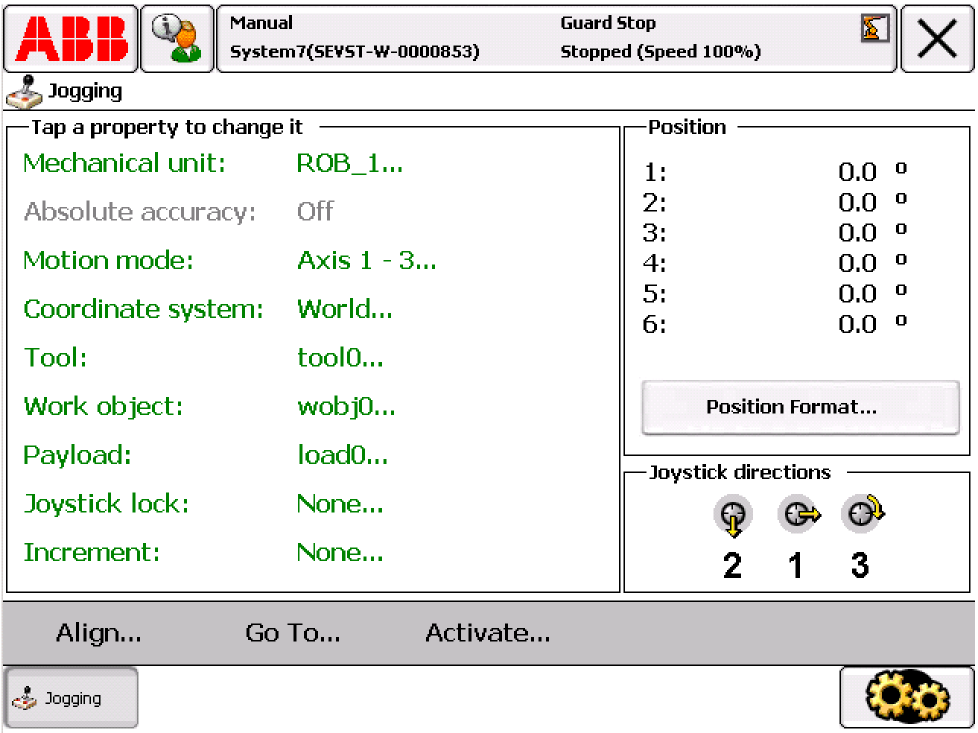
4.3.4. Jogging

Overview

The Jogging functions are found in the Jogging window. The most commonly used are also available under the Quickset menu.

Jogging menu

The illustration shows the functions available under the Jogging menu:



en0400000654

Property/button	Function
Mechanical unit	Select active mechanical unit, described in section *Selecting mechanical unit on page 108*.
Absolute accuracy	**Absolute Accuracy: Off** is default. If the robot has the *Absolute accuracy* option, then **Absolute Accuracy: On** is displayed.
Motion mode	Select motion mode, described in section *Selecting motion mode on page 110*.
Coordinate system	Select coordinate system, e.g. described in section *Jog in base coordinates on page 116*.
Tool	Select tool, described in section *Selecting tool, work object, and payload on page 112*.
Work object	Select work object, described in section *Selecting tool, work object, and payload on page 112*.
Payload	Select payload, described in section *Selecting tool, work object, and payload on page 112*.

Continues on next page

Property/button	Function
Joystick lock	Select locking joystick directions, described in section *Locking the joystick in specific directions on page 122*.
Increment	Select movement increments, described in section *Incremental movement for precise positioning on page 124*.
Position	Displays each axis position in relation to the selected coordinate system.
Position format	Select position format, described in section *How to read the exact position on page 126*.
Joystick directions	Displays current joystick directions, depending on setting in Motion mode. See section *Selecting motion mode on page 110*.
Align...	Align the current tool to a coordinate system. See section *Aligning tools on page 187*.
Go To...	Move the robot to a selected position/target. See section *Moving the robot to a programmed position on page 186*.
Activate...	Activate a mechanical unit. See section *Activating mechanical units on page 195*.

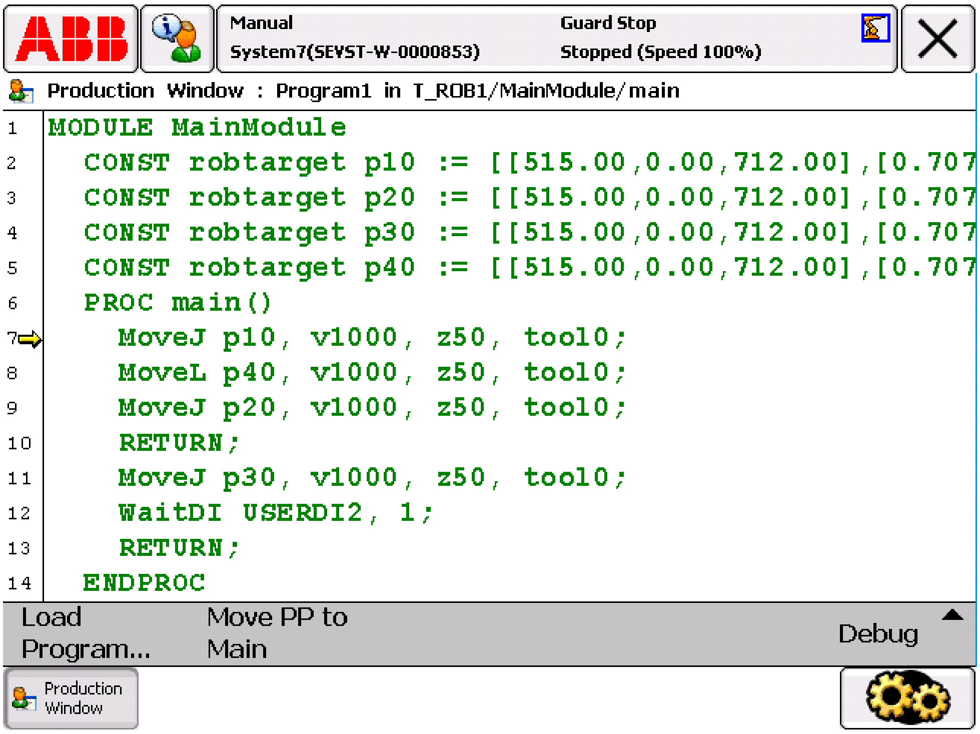
4.3.5. Production window

Overview

The Production window is used to view the program code while the program is running.

Illustration of the Production window

This section illustrates the Production window.



en0400000955

Production window parts and functions

This section describes the parts and functions in the Production window.

Use...	to...
Load Program...	load a new program. Note that the loaded program will not be saved.
Move PP to Main	move the program pointer to the routine main.
Debug	modify position, go to program pointer or motion pointer, or to open the Program editor. How to use the Program editor is described in section *Program editor*, e.g. *Handling of instructions on page 176*, or *Handling of routines on page 172*. Debug is only available in manual mode.

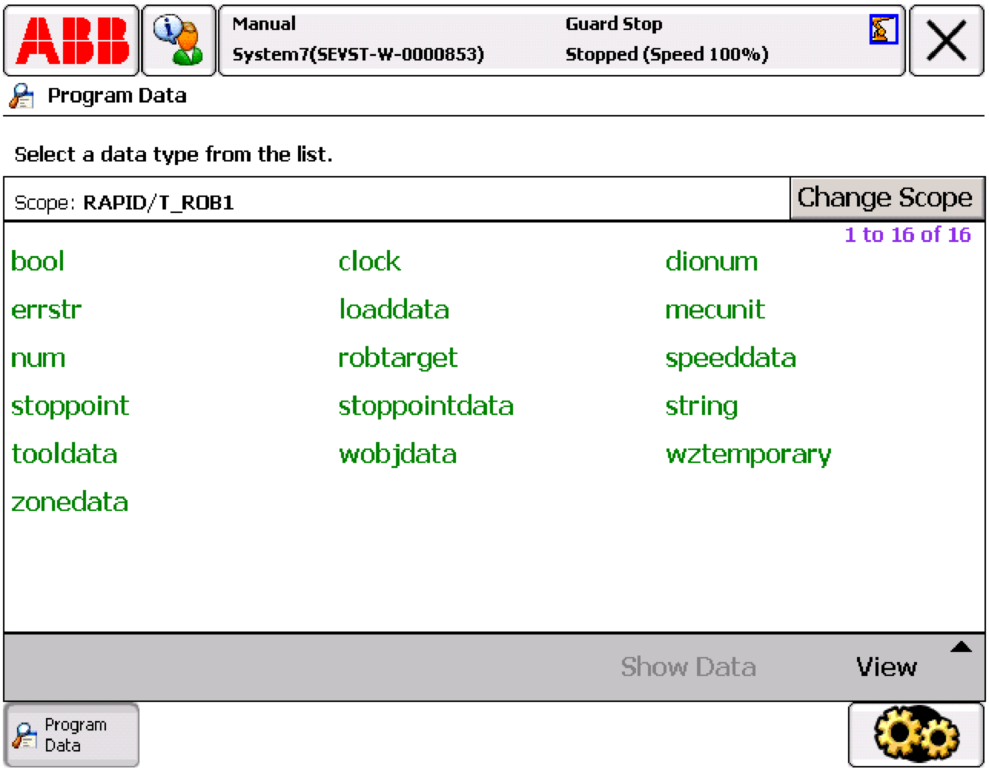
4.3.6. Program data

Overview

The Program data view contains functions for viewing and working with data types and instances. You can open more than one window of the Program data, which can be useful when working with many instances or data types.

Illustration of Program data

This section illustrates the Program data view.



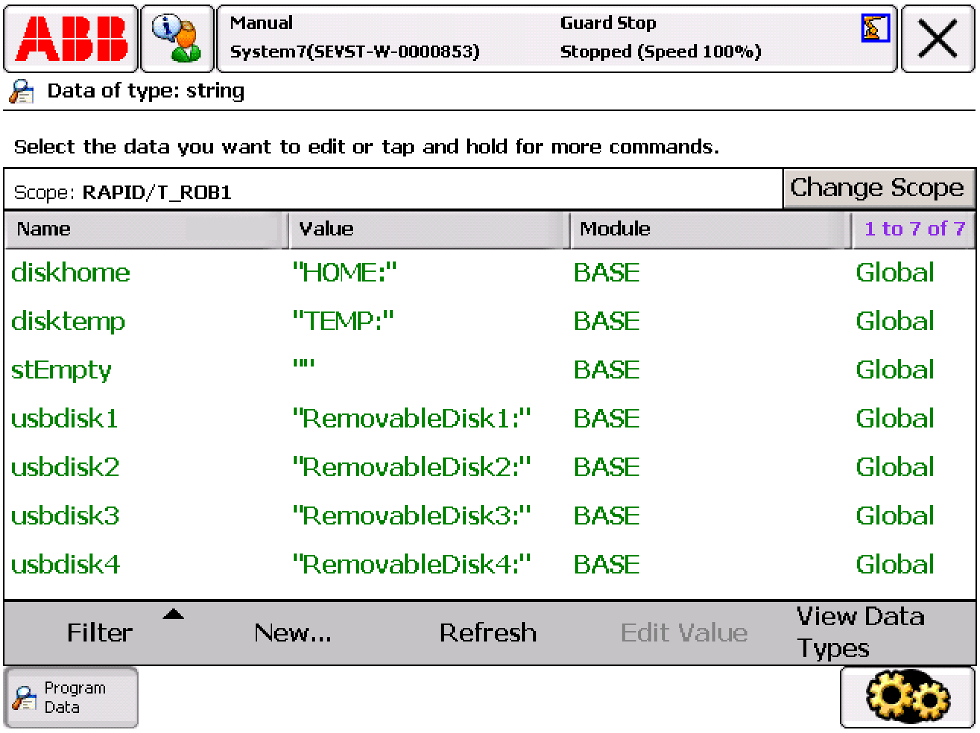
en0400000659

Change scope	changes scope of data types in the list, see section *Viewing data in specific tasks, modules, or routines on page 136*.
Show data	shows all instances of the selected data type.
View	shows all or only used data types.

Continues on next page

Illustration of a data type instances

This section illustrates a list of instances for a data type.



en0500001571

Filter	filters the instances, see *Filtering data on page 101*.
New	creates a new instance of the selected data type, see *Creating new data instance on page 137*.
Refresh	refreshes the list of instances.
Edit value	edits the selected instance's values, see *Editing data instances on page 139*.
View data types	returns to the Program data menu.

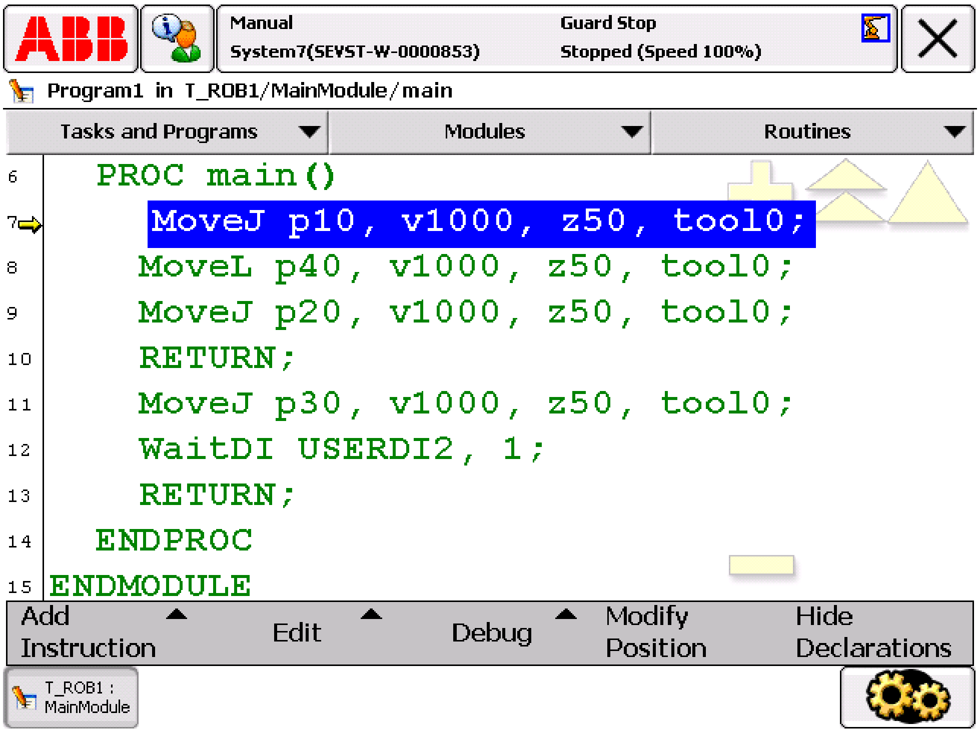
4.3.7. Program editor

Overview

The Program editor is where you create or modify programs. You can open more than one window of the Program editor, which can be useful when working with multitasking programs for instance. The Program editor button in the task bar displays the name of the task.

Illustration of Program editor

This section illustrates the Program editor view.



en0400001143

Tasks and programs	Menu for program operations, see *Handling of programs on page 165*.
Modules	Lists all modules, see *Handling of modules on page 168*.
Routines	Lists all routines, see *Handling of routines on page 172*.
Add instruction	Opens instruction menu, see *Handling of instructions on page 176*.
Edit	Opens edit menu, see *Handling of instructions on page 176*.
Debug	Functions for moving the program pointer, service routines etc, see *Running a service routine on page 209*.
Modify position	See *Modifying positions on page 184*.
Hide declarations	See *Hiding declarations in program code on page 191*.

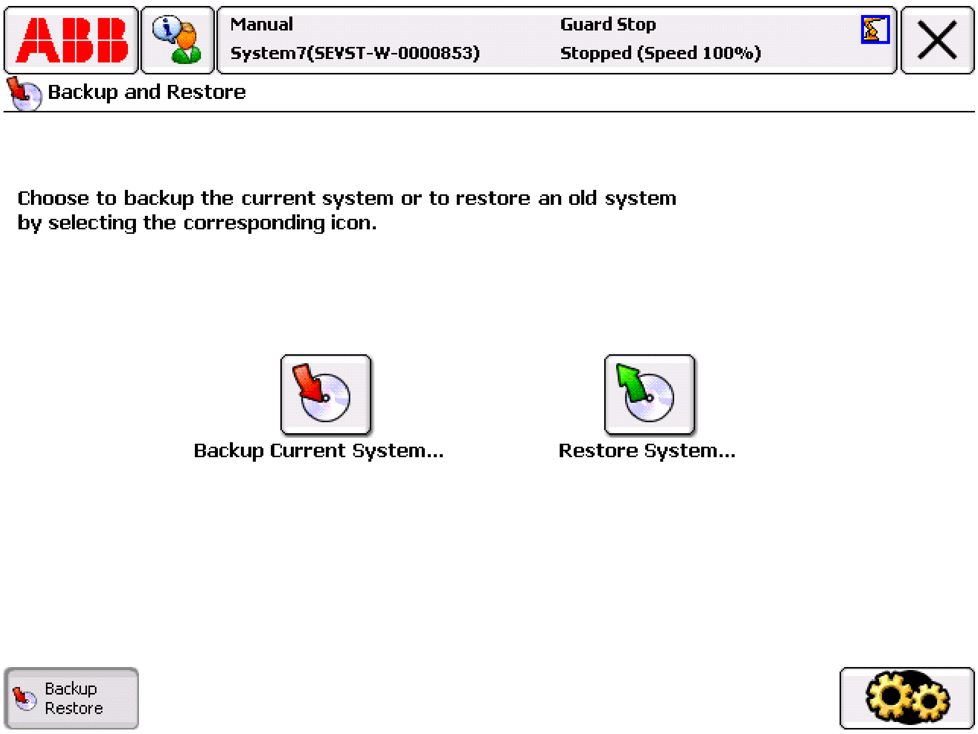
4.3.8. Backup and restore

About backups

The Backup and restore menu is used for performing backups and restoring the system. See section *Backup and restore systems on page 265*.

Illustration of backup and restore

This is the Backup and restore menu.



xx0300000440

Backup current system	See *Backup the system on page 267*.
Restore system	See *Restore the system on page 268*.

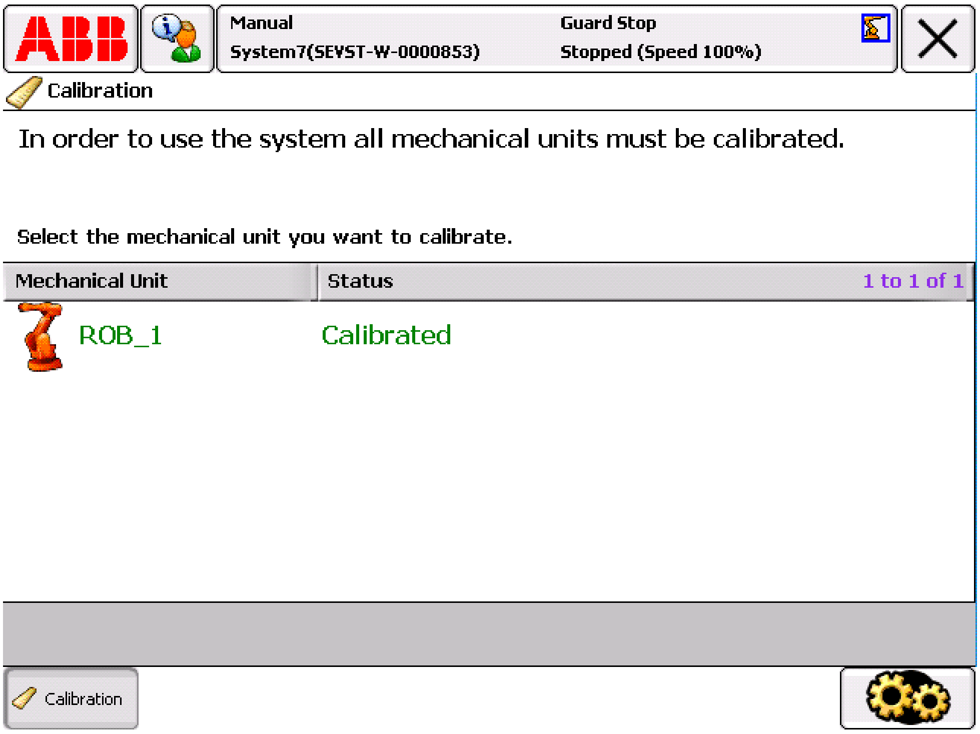
4.3.9. Calibration

About calibration

The Calibration menu is used to calibrate mechanical units in the robot system. Calibration can be performed using the options *Pendulum Calibration* or *Levelmeter Calibration (alternative method)*. See respective instruction manual.

Illustration of Calibration menu

This illustration shows the Calibration menu. All mechanical units are listed and their calibration status is displayed in the Status column.

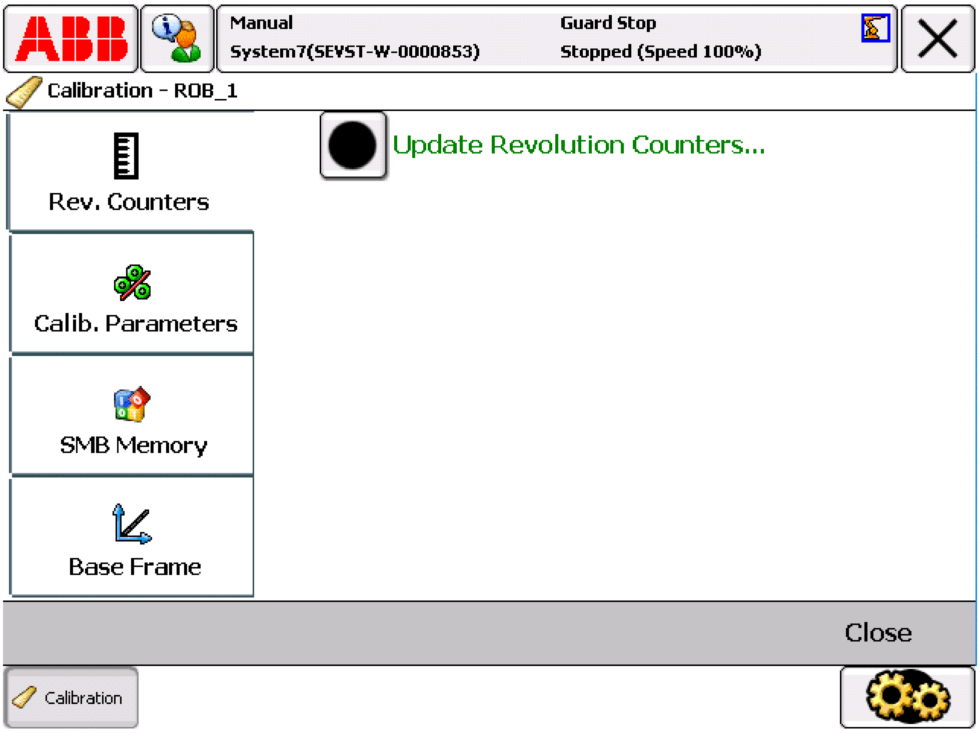


en0400001146

Continues on next page

Calibration menu options

This illustration shows the Calibration menu options after selecting mechanical unit.



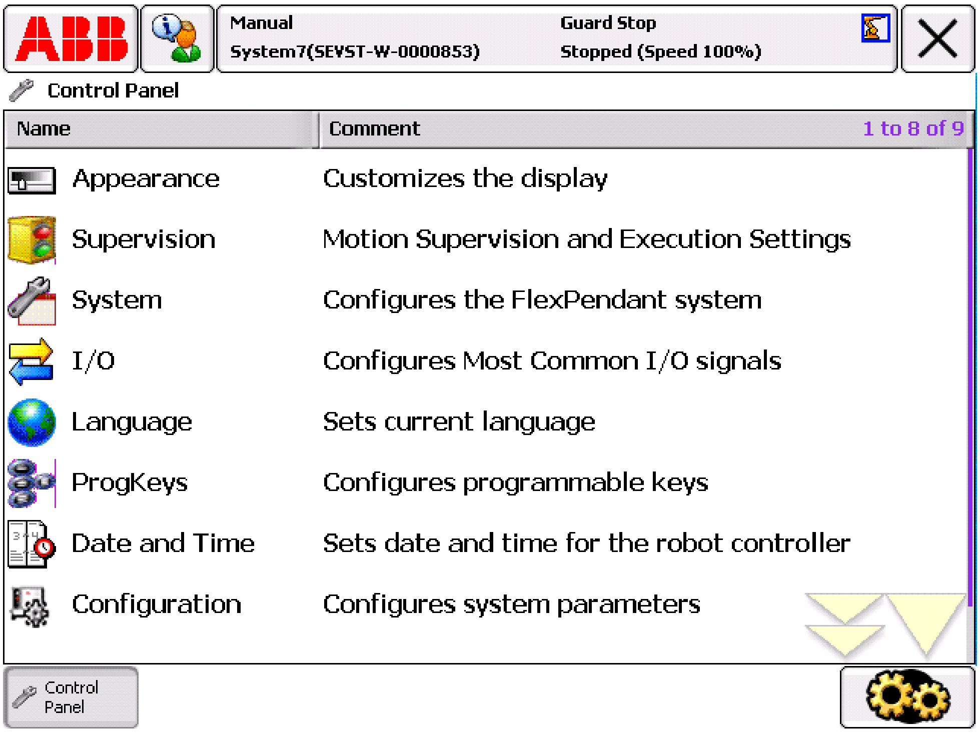
en0400000771

Revolution Counters	See section *Updating revolution counters on page 274*.
Calibration Parameters	See sections *Loading calibration data using the FlexPendant on page 276*, *Editing motor calibration offset on page 277*, and *Fine calibration procedure on FlexPendant on page 278*.
SMB Memory	See section *Serial Measurement Board memory on page 280*.
Base Frame	See section *4 points XZ calibration on page 283*.

4.3.10. Control panel

Control panel

The Control panel contains functions for customizing the robot system and the FlexPendant.



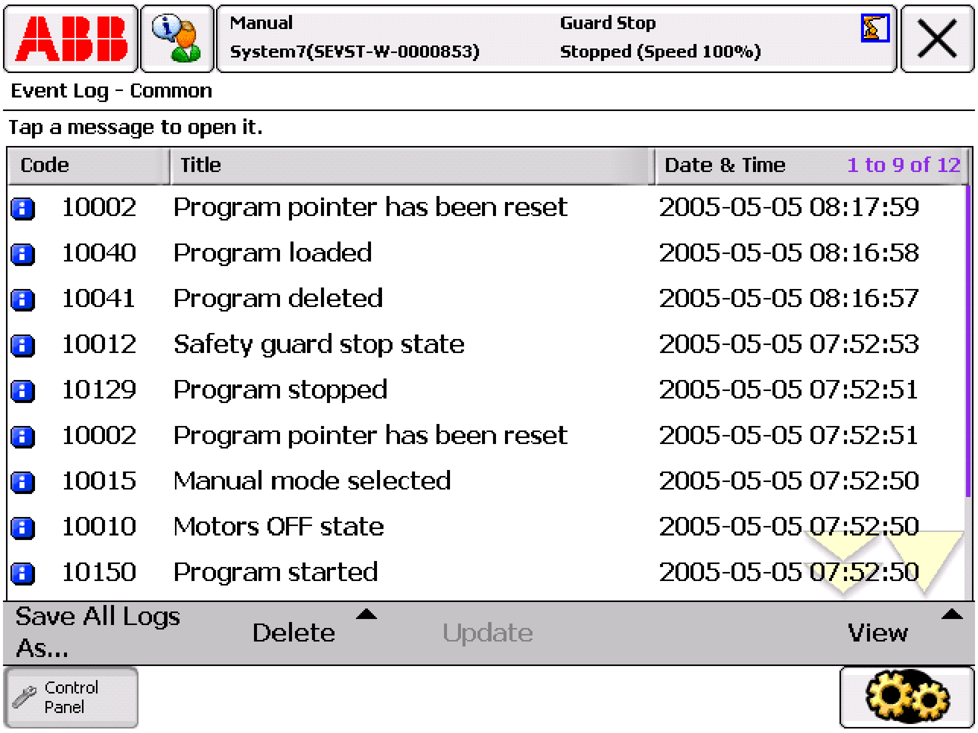
en0400000914

Appearance	Settings to customize the display's brightness and contrast. See *Changing brightness and contrast on page 285*.
Configuration	Configuration of the system parameters configuration. See *Configuring system parameters on page 270*.
Date and Time	Settings for date and time for the robot controller. See *Changing date and time on page 288*.
I/O	Settings for configuring the Most Common I/O list. See *Configuring Most Common I/O on page 289*.
Language	Settings for current language for the robot controller. See *Changing language on page 290*.
ProgKeys	Settings for the four programmable keys on the FlexPendant. See *Editing programmable keys on page 291*.
Supervision	Settings for motion supervision and execution settings. See *Editing supervision settings on page 293*.
System	Configuration of views for operating mode switch and UAS, User Authorization System. See *Configuring view settings on page 295*.
Touch Screen	Recalibration settings for the touch screen. See *Calibrating the touch screen on page 298*.

4.3.11. Event log menu

Event log

The table is a brief summary of all actions that may be performed with the event log.



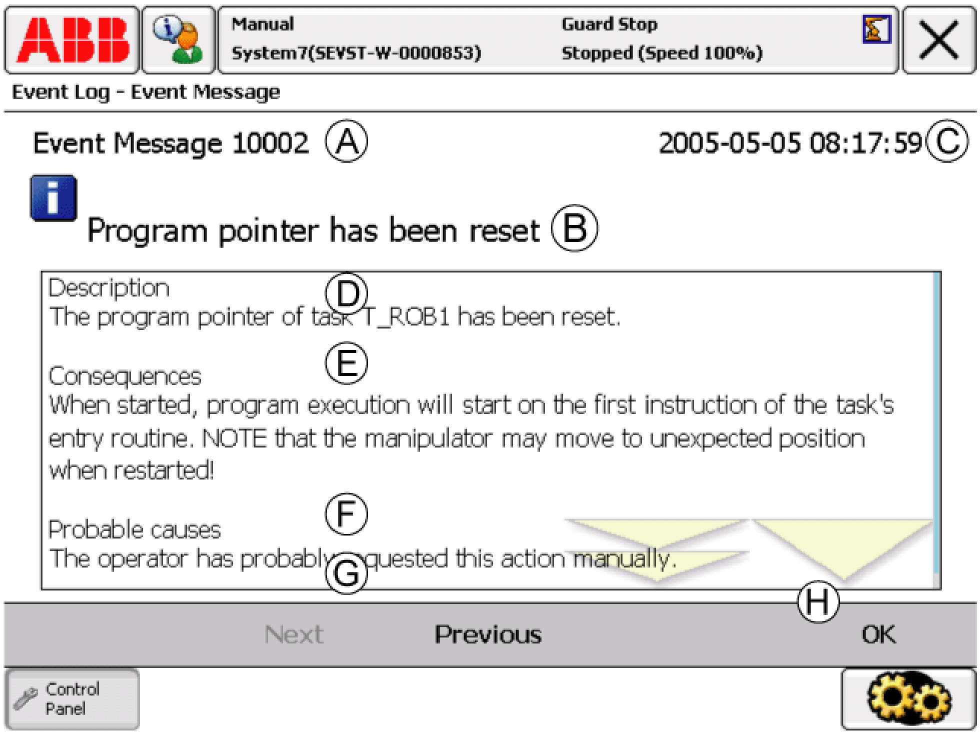
xx0300000447

Actions	Descriptions
The log may be opened.	How to do this is detailed in section *Accessing the event log on page 243*.
A specific message may be viewed.	How to do this is detailed in section *View messages*.
If the log contents do not fit into a single screen, it may be scrolled and/or zoomed.	How to do this is detailed in section *Scrolling and zooming on the FlexPendant on page 100*.
The log may be deleted.	How to do this is detailed in section *Deleting log entries on page 244*.
The log may be saved.	How to do this is detailed in section *Saving log entries on page 245*.
The log may be closed.	How to do this is detailed in section *Accessing the event log on page 243*.

Continues on next page

An event log message

Each event log entry consists of a message describing the event in detail, and it often contains advice on how to solve the problem.



en0300000454

A	Event number. All errors are listed by numbers.
B	Event title. Briefly states what has happened.
C	Event time marker. Specifies exactly when the event occurred.
D	Description. A brief description of the event. Intended to assist in understanding the causes and implications of the event.
E	Consequences. A brief description of any consequences inflicted on the system, transition to other operation mode, emergency stop, caused by the particular event. Intended to assist in understanding the causes and implications of the event.
F	Probable causes. A list of probable causes, listed in order of probability.
G	Recommended actions. A list of the recommended correcting actions, based on the “Probable causes” specified above. These may range from “Replace the xx...” to “Run test program xx...”, i.e. may be actions to isolate the problem as well as fixing it.
H	Acknowledge or OK button.

Related information about logs

Event log messages and more information about the event log are described in *Trouble shooting manual - IRC5*.

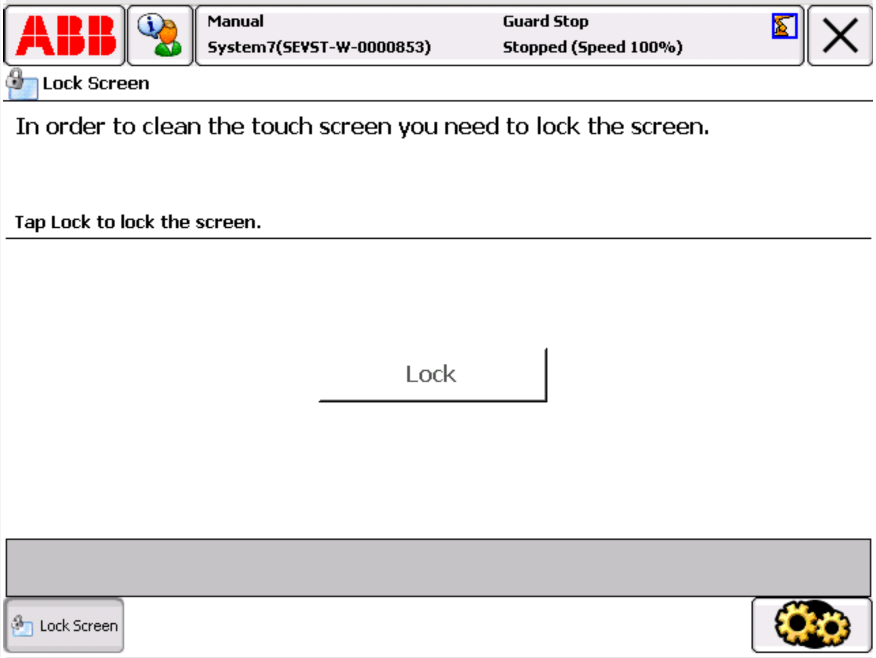
4.3.12. Lock the screen

Overview

The FlexPendant screen may be locked to prevent accidental interference, when e.g. cleaning the screen. How to clean the screen and when to do it is detailed in the *Product Manual - IRC5*.

Locking the screen

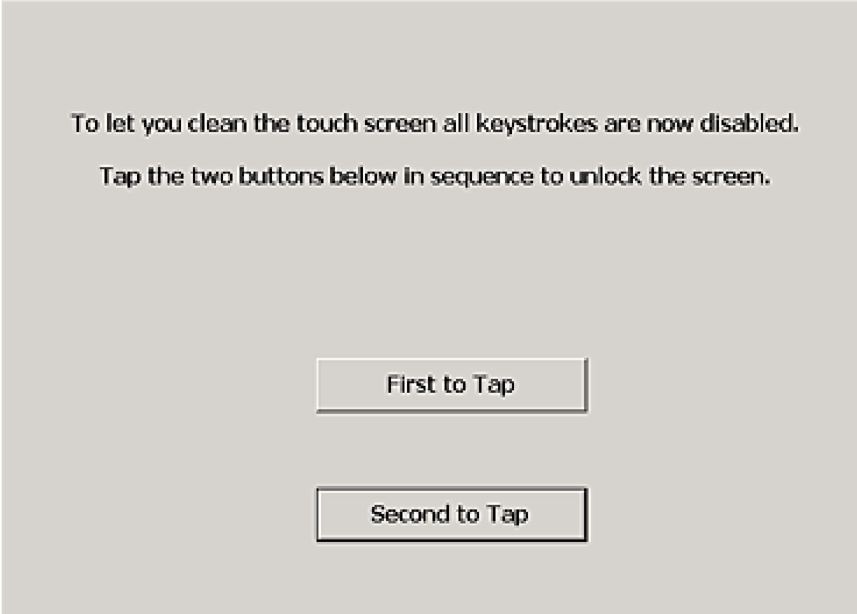
This section describes how to lock the FlexPendant's touch screen.

Step	Action
1.	On the **ABB** menu, tap **Lock Screen**. The following screen is presented:  en0400000657

Continues on next page

4.3.12. Lock the screen

Continued

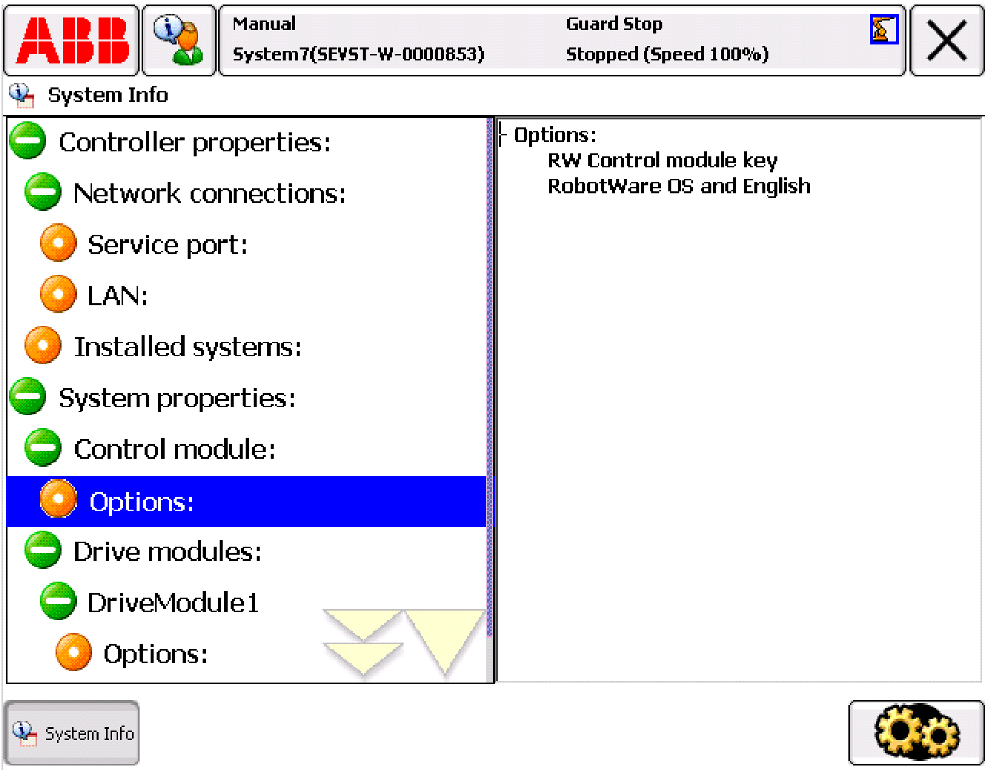
Step	Action
2.	Tap **Lock**. The following screen is presented:  en0400000658
3.	Tapping the two buttons in the correct sequence as specified will unlock the screen.

4.3.13. System info

System info

System info displays all settings, properties and program versions, valid for the controller and the installed system.

Illustration of System info view



en0400000968

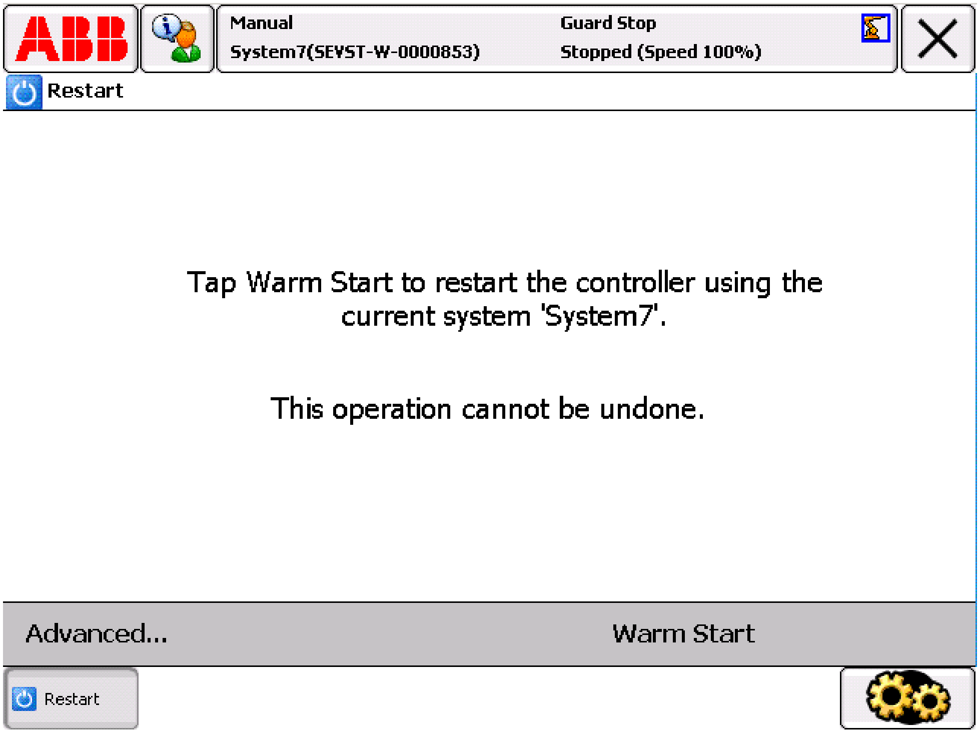
Controller properties	Name of the controller.
Network connections	Settings for the service port and LAN.
Installed systems	Information on the installed systems.
System properties	information about the loaded system.
Control module	Name and key for the control module.
Options	All installed RobotWare options and languages.
Drive module	Lists all drive modules.
Drive module x	Name and key for drive module x.
Options	Options for drive module x, with type of robot etc.
Additional options	Any additional installed options.

4.3.14. Restart

Restart

A running system normally does not need to be restarted.

Tap the **ABB** menu and then **Restart** to restart the system.



en0500001557

Related information

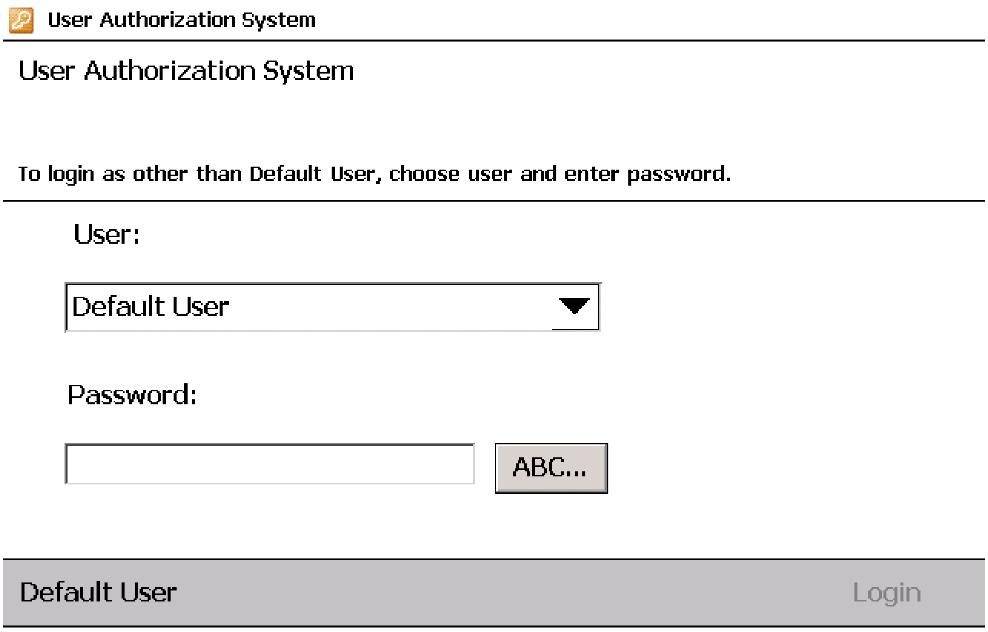
Restart overview on page 253.

4.3.15. Logout

The Logout menu

This section details the Logout menu. More about using this menu is described in *How to logout and login on page 103*.

Logout is availble under the ABB menu.



en0400000947

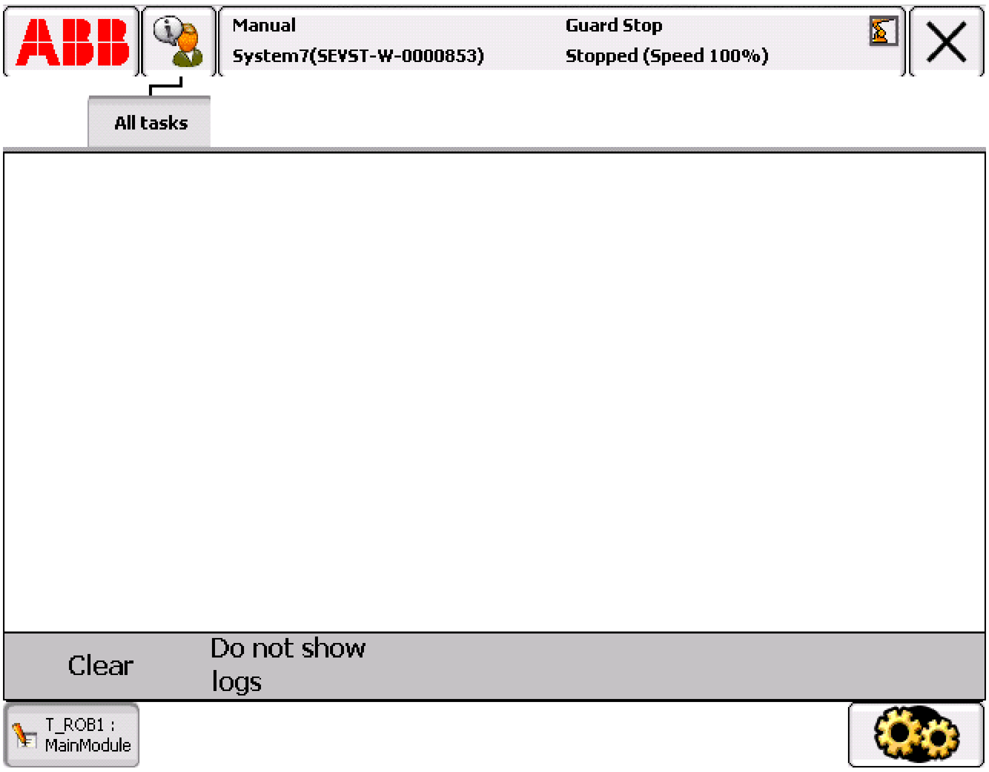
4.4 The status bar menu

4.4.1. Operator window

Operator window

The operator window displays messages from the program. In a multitasking system, all task's messages are displayed in the same operator window. If a message requires action then a separate window for that task will be displayed.

The operator window is opened by tapping the icon to the right of the ABB logo in the status bar. The illustration shows an example of an operator window:



en0400000975

Clear	Clears all messages
Do not show logs	Hides all messages

Messages are written by the program author in RAPID. It can sometimes be useful to hide all messages since this window otherwise will pop up and take focus for each message.

4.4.2. Status bar

Illustration of status bar

The Status bar displays information about the current status, such as operational mode, system, and active mechanical unit.

en0300000490

Part	Name
A	Operator window
B	Operating mode
C	System name (and controller name)
D	Controller state
E	Program state
F	Mechanical units. The selected unit (and any unit coordinated with the selected) is marked with a border. Active units are displayed in color, while deactivated units are grey.

4.5 The QuickSet menu

4.5.1. The Quickset menu

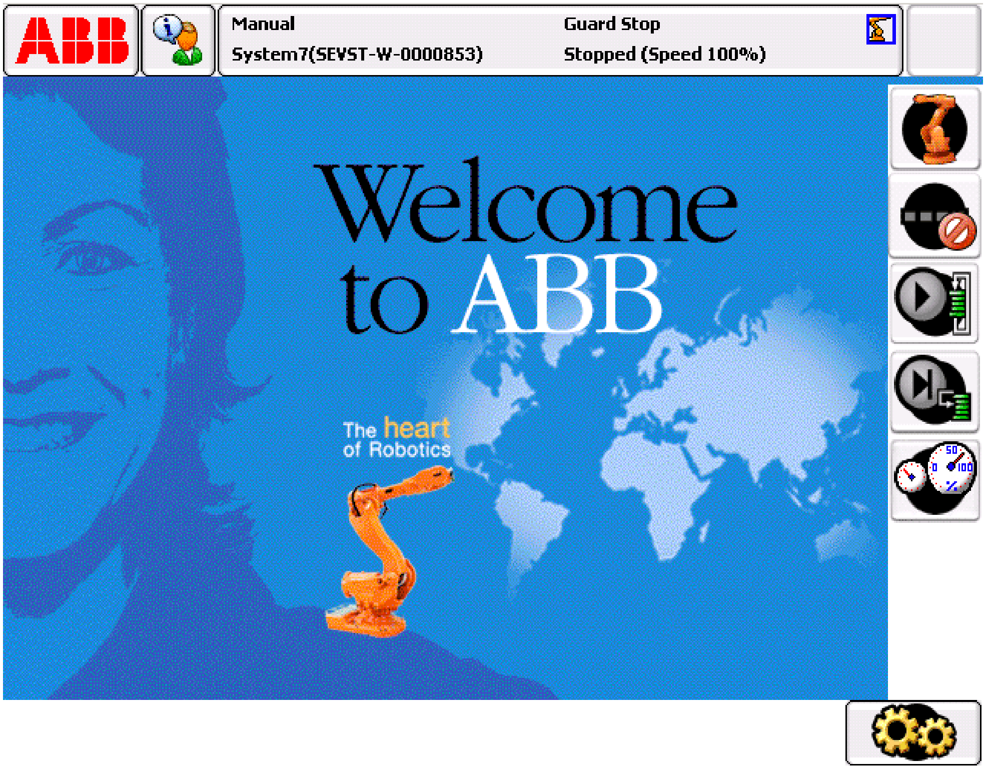
General

The QuickSet menu provides a quicker way to change among other things jog properties rather than using the **Jogging** view.

Each item of the menu uses a symbol to display the currently selected property value or setting. Tap the Quickset button to display available property values.

Illustration of the Quickset menu

This section describes the buttons in the Quickset menu.



en0300000471

	Mechanical unit, see section *Quickset menu, Mechanical unit on page 128*
	Increment, see section *Quickset menu, Increment on page 132*
	Run Mode, see section *Quickset menu, Run Mode on page 201*
	Step Mode, see section *Quickset menu, Step Mode on page 203*
	Speed, see section *Quickset menu, Speed on page 208*

4.6 Basic procedures

4.6.1. Using the soft keyboard

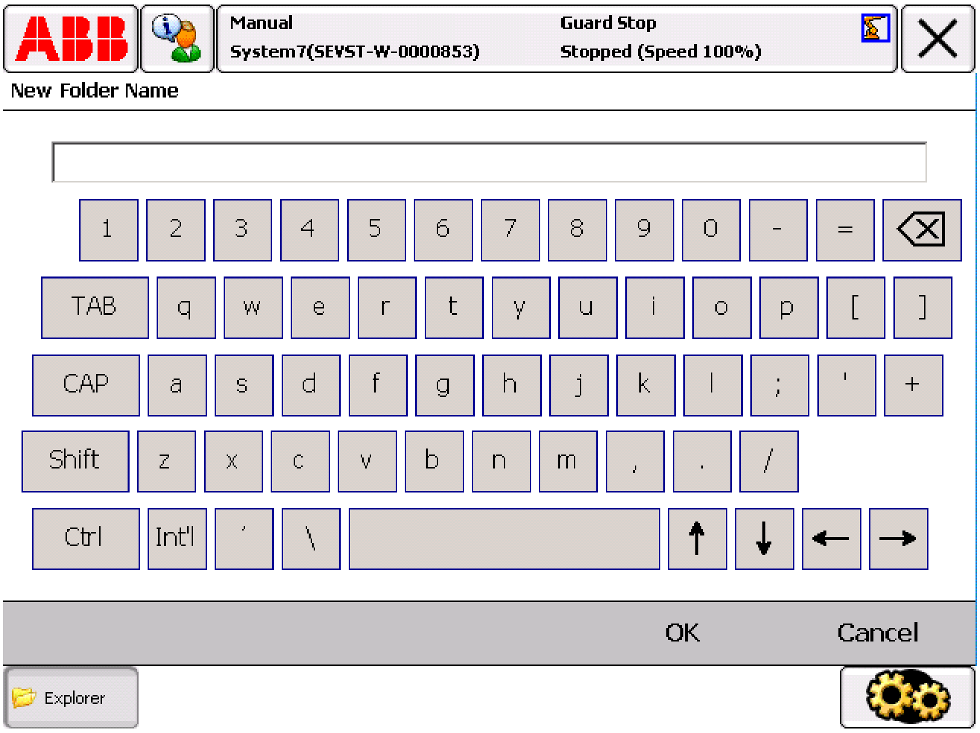
Using the soft keyboard

The soft keyboard is used frequently when operating the system, e.g. when entering file names or parameter values.

The soft keyboard works as an ordinary keyboard with which you can place the insertion point, type and correct typing errors. Tap letters, numbers and special characters to enter your text or values.

Illustration soft keyboard

This illustration shows the soft keyboard on the FlexPendant.



en0300000491

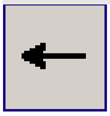
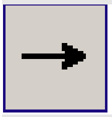
Using international characters

All western characters can be used, also in usernames and passwords. To access international characters, tap the **Int'l** button on the soft keyboard.

Continues on next page

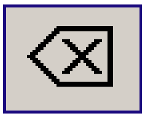
How to change the insertion point

Tap the arrow keys to change the insertion point, for instance when correcting typing errors.

If you need to move...	then tap...
backward	 xx0300000492
forward	 xx0300000493

How to delete

1. Tap the **Backspace** key (top right) to delete characters to the left of the insertion point.



xx0300000494

4.6.2. Messages on the FlexPendant

Overview of messages

The FlexPendant displays messages from the system. These can be status messages, error messages, program messages, or requests for action from the user. Some require actions, and some are plain information.

Event log messages

The event log messages are messages from the RobotWare system about system status, events, or errors.

How to work with the event log messages is described in section *Handling the event log on page 243*. All messages are also described in the manual *Trouble shooting - IRC5*.

System messages

Some messages sent out by the system are not from the event log. They can come from other applications, such as RobotStudio^{Online}.

To be able to change configurations and settings in the system from RobotStudio^{Online}, the user must request write access. This generates a message on the FlexPendant where the operator can grant or deny access. The operator can at any time decide to withdraw the write access.

How to request access and work with RobotStudio^{Online} is described in *Operator's manual - RobotStudio Online*.

Program messages

RAPID programs can send out messages to the Operator window, see section *Operator window on page 94*.

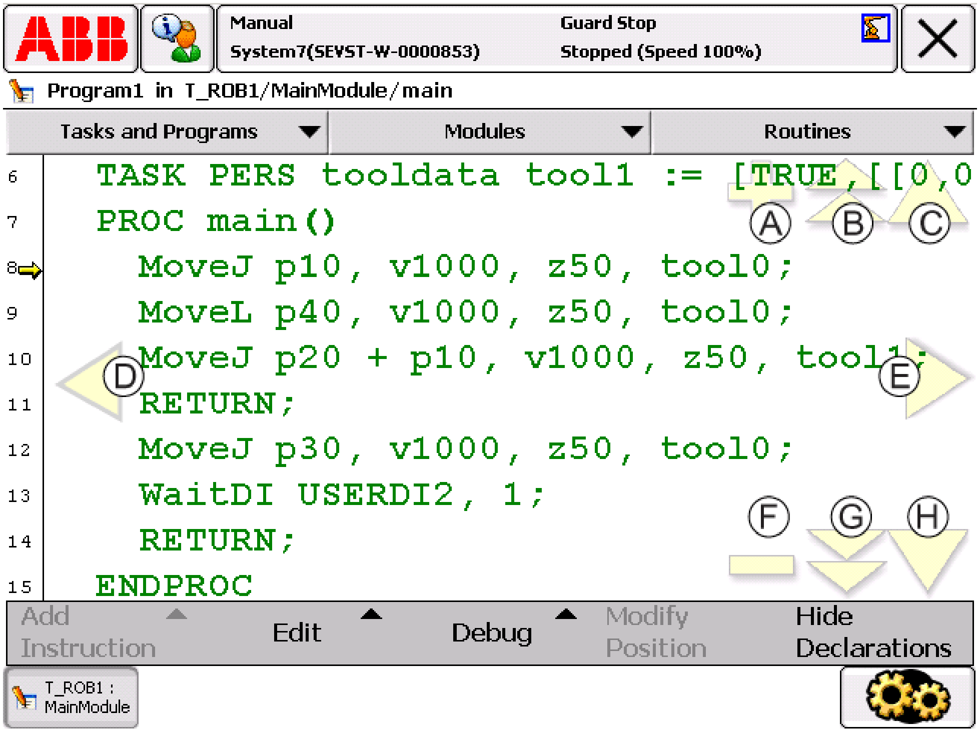
How to generate program messages is described in *RAPID reference manual - Instructions*.

4.6.3. Scrolling and zooming on the FlexPendant

Overview

The entire contents of a screen may not be visible at the same time. To see the entire contents, you may:

- Scroll up/down (and sometimes left/right)
- Zoom in or out (only available in the Program editor)



en0400000685

A	Zoom in (larger text)
B	Scroll up (the height of one *page*)
C	Scroll up (the height of one *line*)
D	Scroll left
E	Scroll right
F	Zoom out (smaller text)
G	Scroll down (the height of one *page*)
H	Scroll down (the height of one *line*)

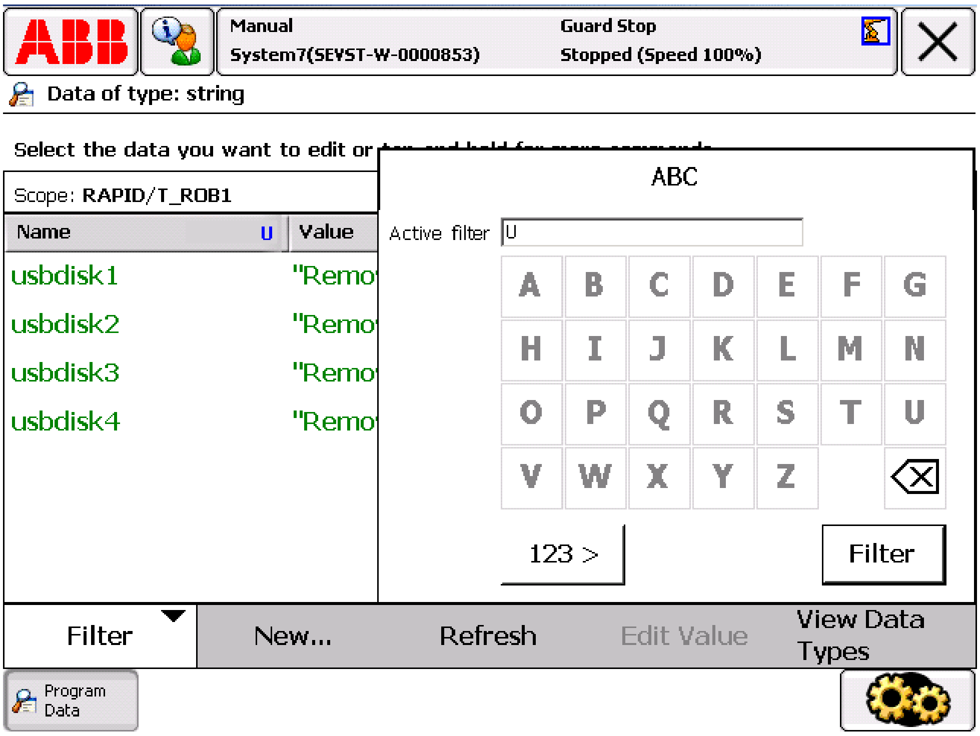
4.6.4. Filtering data

Filtering data

In many of the FlexPendant’s menues you can filter data. This is useful e.g. when looking at instances of data types where there can be more available than is possible to overlook. By filtering out instances starting with a specific character, the number of choices can be reduced.

Illustration of filtering function

The filter function is switched on until the active filter is removed (using the keyboard).



en0500001539

Filter	Open or close the filtering function
Active filter	Displays the current filter. This is also displayed in the top of the list of items
123 / ABC	Switch between numeric or alphabetic keyboard
Filter	Applies the filter

4.6.5. Process applications

Process applications

Custom process applications are started from the ABB menu. Each application is listed as a menu item together with the FlexPendant's views.

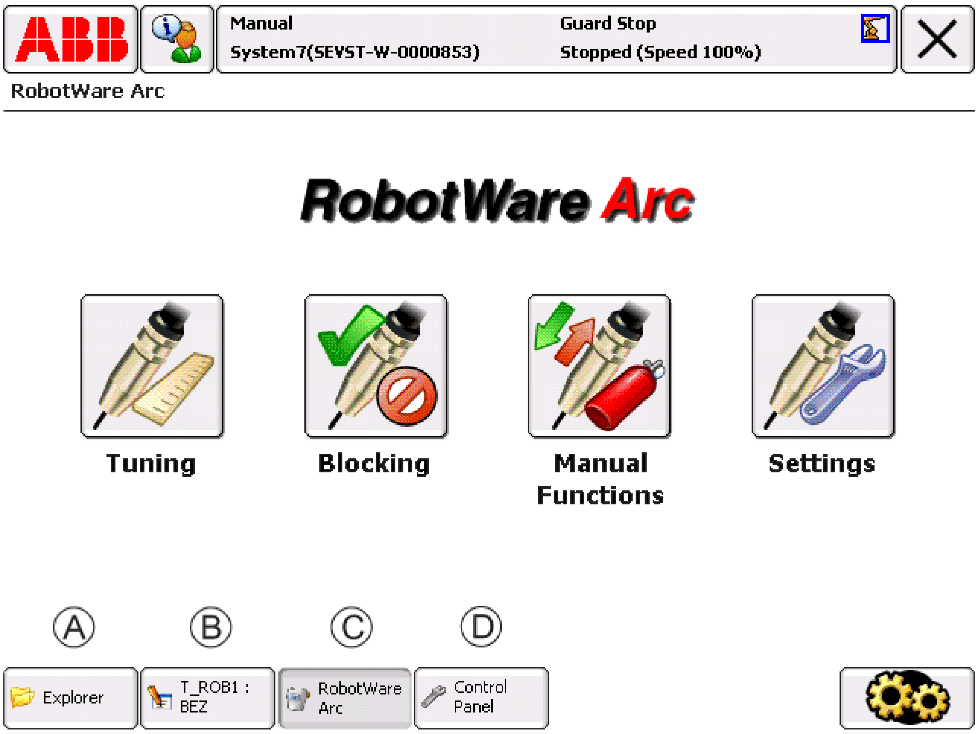
Start a process application

This procedure describes how to start a process application.

Step	Action
1.	Tap the **ABB** button to display the ABB menu. Process applications are listed in the menu.
2.	Tap the name of the process application to start.

Switch between started process applications

A started application has a quick-button in the taskbar, just like FlexPendant views. Tap the buttons to switch between the started applications and views.



en0400000768

The views and process application running in this case are:

A	FlexPendant Explorer view
B	Program editor view
C	RobotWare Arc, a process application
D	Control panel view

4.6.6. How to logout and login

Logout procedure

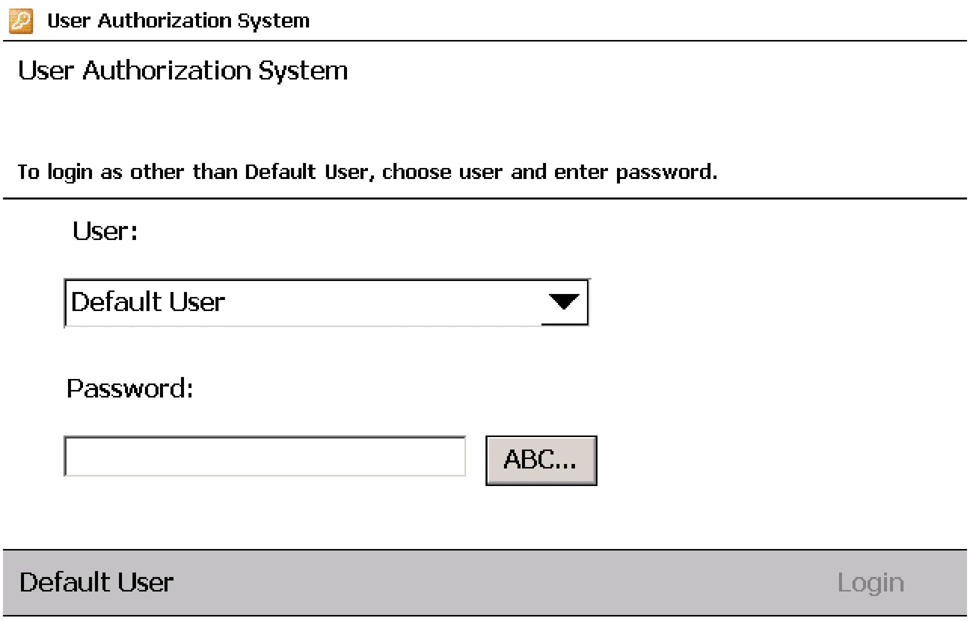
The procedure details how to logout of the system.

Step	Action
1.	Tap **Logout** on the ABB menu.
2.	Tap **Yes** on the logout question.

Login procedure

The procedure details how to login to the controller, using the User Authorization System, UAS. UAS can limit the available functions for users.

After a logout, the Login window automatically appears.



en0400000947

Step	Action	Info
1.	Tap on the **User** menu to choose user. If you tap **Default User**, no password is required, and you are logged in automatically.	If the user you have chosen has a password you must use the soft keyboard to enter password.
2.	Tap **ABC...** to display the soft keyboard. After entering the password tap **OK**.	
3.	Tap **Login**.	

Continues on next page

Handling users and authorization levels

Read more on how to add users or set the authorization in *Operator's manual - RobotStudio Online*.

How to edit what views or functions are hidden for certain users is described in *Configuring view settings on page 295*.

5 Jogging

5.1. Introduction to jogging

What is jogging?

To jog is to manually position or move robots or external axes using the FlexPendant joystick.

When can I jog?

You jog in manual mode. Jogging is possible regardless of what view is displayed on the FlexPendant, however you cannot jog during program execution.

About motion modes and robots

The selected motion mode and/or coordinate system determines the way the robot moves.

In linear motion mode the tool center point moves along straight lines in space, in a “move from point A to point B” fashion. The tool center point moves in the direction of the selected coordinate system’s axes.

Axis-by-axis mode moves one robot axis at a time. It is then hard to predict how the tool center point moves.

About motion modes and additional axes

Additional axes can only be jogged axis-by-axis. An additional axis can either be designed for some kind of linear motion or for rotational (angular) motion. Linear motion is used in conveyers, rotational motion in many kinds of workpiece handlers.

Additional axes are not affected by the selected coordinate system.

About coordinate systems

Positioning a pin in a hole with a gripper tool can be performed very easily in the tool coordinate system, if one of the coordinates in that system is parallel to the hole. Performing the same task in the base coordinate system may require jogging in both x, y, and z coordinates, making precision much more difficult.

To select the proper coordinate systems to jog in will make jogging easier but there are no simple or single answers to which coordinate system to choose.

A certain coordinate system will make it possible to move the tool center point to the target position with fewer joystick moves than another.

Conditions such as space limitations, obstacles or the physical size of a work object or tool will also guide you to the proper judgement.

Read more about coordinate systems in section *What is a coordinate system? on page 305*.

5.2 Jogging concept

5.2.1. Restrictions to jogging

Jog additional axes

Additional axes can only be jogged axis-by-axis. Please see *Application manual - Additional axes*.

Jog mechanical units that are not calibrated

An uncalibrated mechanical unit can only be jogged axis-by-axis. Its working range will not be checked.

If the mechanical unit is not calibrated the text “Unit not calibrated” will be displayed in the **Position** area of the **Jogging** window.

CAUTION!

Mechanical units whose working range is not controlled by the robot system can be moved to dangerous positions. Mechanical stops should be used and configured to avoid danger to equipment or personnel.

Jog robot axes in independent mode

It is not possible to jog axes in independent mode. You need to return the axes to normal mode in order to jog. Please see *Application manual - Motion functions and events* for details.

Jog while using world zones

With the option World Zones installed, defined zones will restrict motion while you jog. Please see *Application manual - Motion functions and events* for details.

Jog with axis loads not set

If equipment is mounted on any of the robot axes, then axes loads must be set. Otherwise overload errors might occur when jogging.

How to set axis loads are described in the Product Manuals delivered with your robot.

Jog with tool or payload weights not set

If the weight of tools and payloads is not set, then overload errors might occur when jogging. Loads for additional axes controlled by specific software (dynamic models) can only be set in programing.

5.2.2. Coordinated jogging

Coordination

A robot that is coordinated to a work object will follow the movements of that work object.

Coordinated jogging

If the mechanical unit moving the work object is jogged, any robot that is currently coordinated with the work object will move so that it maintains its relative position to the work object.

Set up coordination

Step	Action	Info
1.	Select the robot that is to be coordinated to another mechanical unit.	See *Selecting mechanical unit on page 108*.
2.	Set **Coordinate system** to Work Object.	See *Jog in work object coordinates on page 119*.
3.	Set **Work object** to the work object moved by the other mechanical unit.	See *Selecting tool, work object, and payload on page 112*.
4.	Select the mechanical unit that moves the work object.	Any jogging, while this mechanical unit is selected, will also affect the robot that is coordinated with it.

Coordinating robots

Coordinating robots, so that when jogging one robot another robot will follow, requires the option *MultiMove*.

5.3 Basic settings for jogging

5.3.1. Selecting mechanical unit

Examples of use

Your robot system may consist of more than a single robot. There can also be other mechanical units such as workpiece handlers or additional axes mounted on the robot that can also be jogged.

If your system only has a single robot without additional axes, then you do not need to select mechanical unit.

Identifying mechanical units

Each mechanical unit that can be jogged is represented in the mechanical units list. The name of the unit is defined in the system configuration. Each unit also has a symbol that is used in the Status bar, see section *Status bar on page 95*.

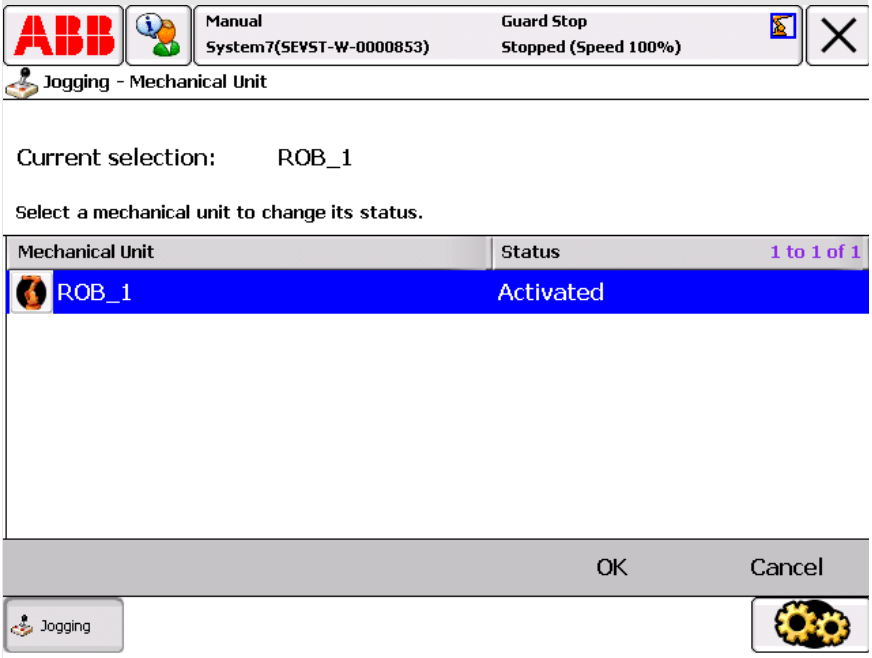
Please consult your plant or cell documentation to see which mechanical units are available in your robot system.

Selecting mechanical unit

This procedure describes how to select a mechanical unit to jog.

Step	Action
1.	On the **ABB** menu, tap **Jogging**.

Continues on next page

Step	Action
2.	Tap **Mechanical Unit**.  en0400000653 A list of available mechanical units is displayed.
3.	Tap the mechanical unit to be jogged, and then tap **OK**. The selected mechanical unit is used until you choose another unit, even if you close the **Jogging** window.

TIP!

Use the **QuickSet** menu to switch between mechanical units faster.

How jogging properties apply

Any changes you make to jogging properties only affects the currently selected mechanical unit.

All jogging properties are saved and restored when you return to jog that mechanical unit.

Related information

Mechanical units can be activated or deactivated using the Activate function in the Jogging menu, see section *Activating mechanical units on page 195*.

5.3.2. Selecting motion mode

Overview

The **Joystick Directions** area shows how joystick axes correspond to the selected coordinate system's axes.

CAUTION!

The Directions properties is not intended to show the direction in which the mechanical unit will move. Always try jogging with small joystick movements so that you learn the true directions of the mechanical unit.

Selecting motion mode

This procedure describes how to select motion mode.

Step	Action	Info
1.	Tap **ABB**, then **Jogging**.	
2.	Tap **Motion mode**.	
3.	Tap on the mode you want and then tap **OK**.	The significance of the joystick directions are shown in **Joystick direction** after making the selection.

Joystick directions

The significance of the joystick directions depends on what motion mode has been selected. The following are available:

Motion mode	Illustration joystick	Description
Linear	 en0400001131	Linear mode is described in section *Setting the tool orientation on page 113*.
Axis 1-3 (default for robots)	 en0300000536	Axis 1-3 mode is described in section *Jog axis by axis on page 114*.

Continues on next page

Motion mode	Illustration joystick	Description
Axis 4-6	 en0300000537	Axis 4-6 mode is described in section *Jog axis by axis on page 114*.
Reorient	 en0400001131	Reorient mode is described in section *Setting the tool orientation on page 113*.

5.3.3. Selecting tool, work object, and payload

Overview

It is always important to choose the proper tool, work object, or payload. It is absolutely vital when you create a program by jogging to the target positions.

Failing to do so will most likely result in overload errors and/or incorrect positioning either when you jog or when you run the program in production.

Selecting tool, work object, and payload

Step	Action
1.	On the **ABB** menu, choose **Jogging** to view jogging properties.
2.	Tap **Tool**, **Work object**, or **Payload** to display the lists of available tools, work objects or payloads.
3.	Tap the tool, work object, or payload of choice followed by **OK**.

5.3.4. Setting the tool orientation

Examples of use

Tools for arc welding, grinding and dispensing must be oriented in a particular angle to the work piece to obtain the best result. You also need to set up the angle for drilling, milling or sawing.

In most cases you set the tool orientation when you have jogged the tool center point to a specific position such as the starting point for a tool operation. After you have set the tool orientation you continue to jog in linear motion to complete the path and the supposed operation.

Definition of tool orientation

The tool orientation is relative to the currently selected coordinate system. From a user perspective however this is not noticeable.

Setting the tool orientation

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Motion Mode**, then tap **Reorient** followed by **OK**.
3.	If not already selected, select the proper tool by following the procedure in *Selecting tool, work object, and payload on page 112*.
4.	Press and hold the enabling device to activate the mechanical unit's motors. Move the joystick and the tool's orientation changes.

TIP!

Use the **QuickSet** menu to select jogging mode faster.

5.3.5. Jog axis by axis

Examples of use

Use axis by axis jogging when you need to:

- move the mechanical unit out of hazardous positions.
- move a robot out of singularities.
- position a robot's axes for calibration.

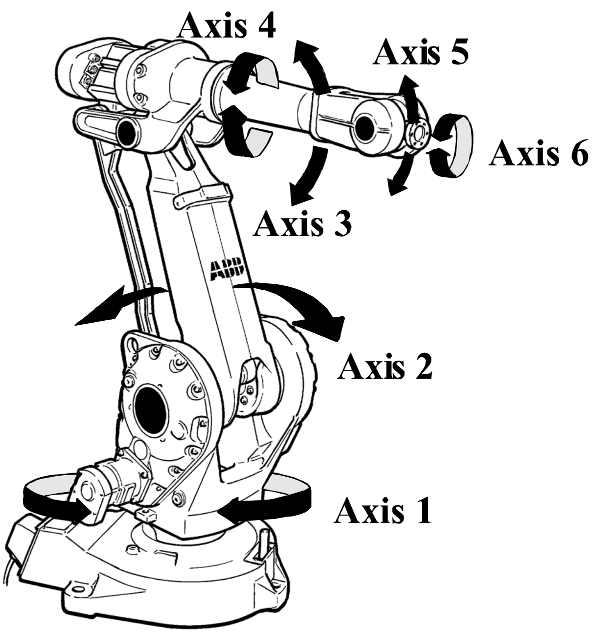
Select axes to move

Step	Action
1.	On the **ABB** menu, tap **Jogging** to view jogging properties.
2.	Tap **Motion Mode**, then select the appropriate axes, see description in *Illustration of axes and joystick directions on page 114*.
3.	Tap **OK** to complete.
4.	Press the enabling device and jog the axes.

Illustration of axes and joystick directions

The regular six axes of a generic manipulator may be jogged manually using the three dimensions of the joystick as specified below. Please check your plant or cell documentation to determine the physical orientation of any additional axes.

The illustration shows the movement patterns of the manipulator's axes.



xx0300000520

Continues on next page

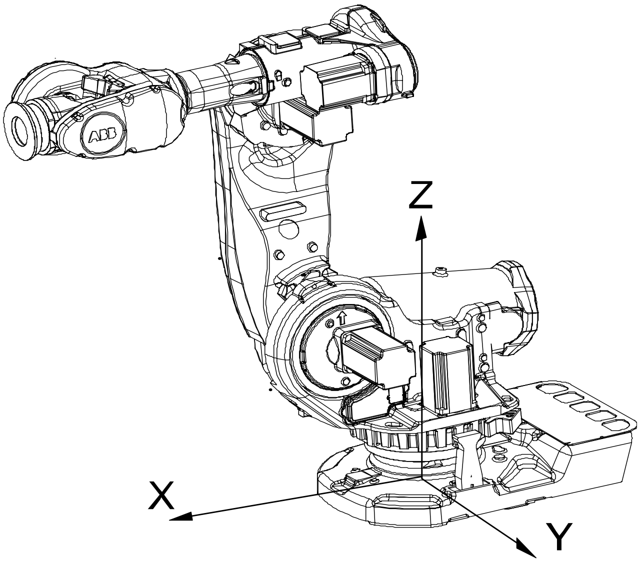
If you want to move…	then tap…	for joystick directions…
axis 1, 2 or 3	 en0300000534	 en0300000536
axis 4, 5 or 6	 en0300000535	 en0300000537

CAUTION!

The orientation of any mounted tool will be affected by this procedure. If the resulting orientation is important, perform the procedure described in *Setting the tool orientation on page 113* when finished.

5.3.6. Jog in base coordinates

Base coordinates definition



xx0300000495

Jog in base coordinates

Step	Action
1.	On the **ABB** menu, tap **Jogging** to view jogging properties.
2.	Tap **Motion Mode**, then tap **Linear** followed by **OK**. You don’t need to perform this step if you previously selected linear motion.
3.	Tap **Coordinate System**, then tap **Base** followed by **OK**.
4.	Press and hold the enabling device to activate the manipulator’s motors.
5.	Move the joystick and the mechanical unit moves along.

TIP!

Use the **QuickSet** menu to select jogging mode faster.

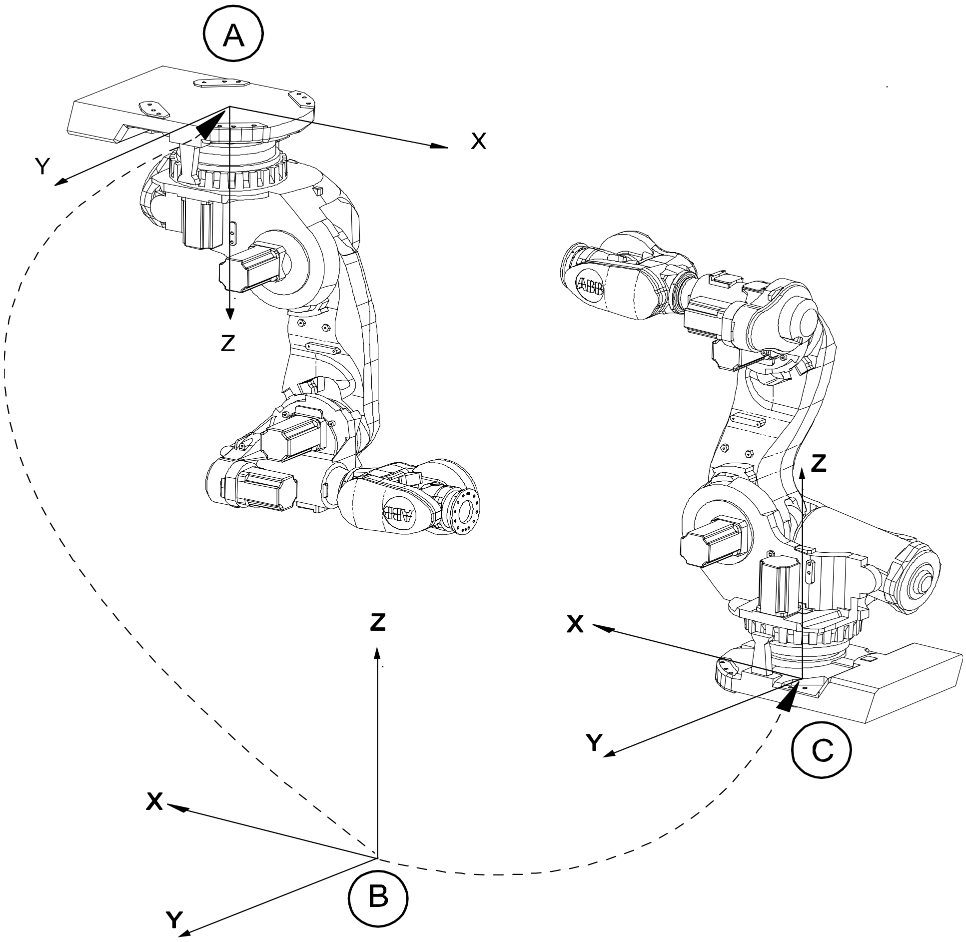
5.3.7. Jog in world coordinates

Examples of use

For example, you have two robots, one floor mounted and one inverted. The base coordinate system for the inverted robot would be upside down.

If you jog in the base coordinate system for the inverted robot, movements will be very difficult to predict. Choose the shared world coordinate system instead.

World coordinates definition



en0300000496

A	Base coordinate system
B	World coordinate system
C	Base coordinate system

Continues on next page

Jog in world coordinates

Step	Action
1.	On the **ABB** menu, tap **Jogging** to view jogging properties.
2.	Tap **Motion Mode**, then tap **Linear** followed by **OK**. You don't need to perform this step if you previously selected linear motion.
3.	Tap **Coordinate system**, then tap **World** followed by **OK**.
4.	Press and hold the enabling device to activate the manipulator's motors.
5.	Move the joystick and the mechanical unit moves along.

TIP!

Use the **QuickSet** menu to select jogging mode faster.

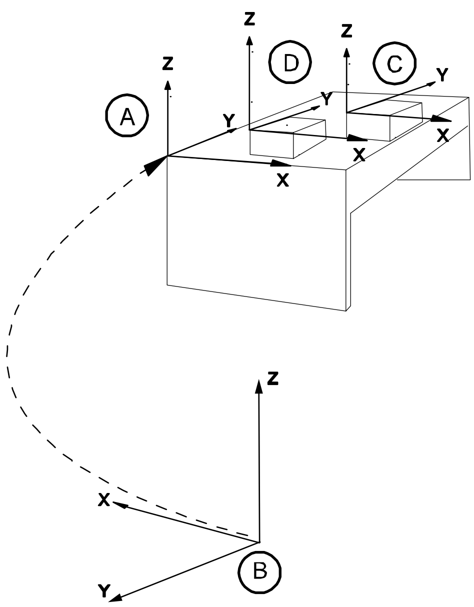
5.3.8. Jog in work object coordinates

Examples of use

For example, you are determining the positions of a number of holes to be drilled along the edge of the work object.

You are creating a weld between two walls in a box.

Work object coordinates definition



en0300000498

A	User coordinate system
B	World frame
C	Work object coordinate system
D	Work object coordinate system

Jog in work object coordinates

Step	Action
1.	On the **ABB** menu, tap **Jogging** to view jogging properties.
2.	Tap **Motion Mode**, then tap **Linear** followed by **OK**. This is not required if you previously selected linear motion.
3.	Tap **Work object** to select work object..
4.	Tap **Coordinate system**, then tap **Work Object** followed by **OK**.
5.	Press and hold the enabling device to activate the manipulator’s motors.
6.	Move the joystick and the mechanical unit moves along.

TIP!

Use the **QuickSet** menu to select jogging mode faster.

5.3.9. Jog in tool coordinates

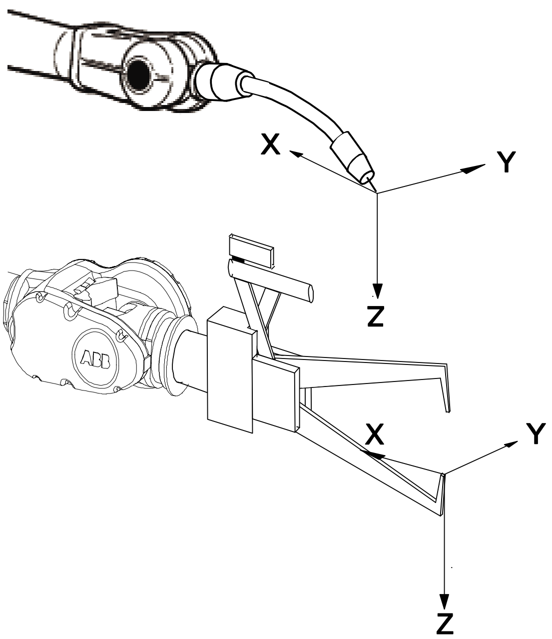
Examples of use

Use the tool coordinate system when you need to program or adjust operations for threading, drilling, milling or sawing.

Tool coordinates definition

The tool coordinate system has its zero position at the center point of the tool. It thereby defines the position and orientation of the tool. The tool coordinate system is often abbreviated to TCP (Tool Center Point) or TCPF (Tool Center Point Frame).

When jogging a robot the tool coordinate system is useful when you don't want to change the orientation of the tool during the movement, for instance moving a saw blade without bending it.



en0300000497

Jog in tool coordinates

Step	Action
1.	On the **ABB** menu, tap **Jogging** to view jogging properties.
2.	Tap **Motion mode**, then tap **Linear** followed by **OK**. You do not need to perform this step if you previously selected linear motion.
3.	Select the proper tool, and if using a stationary tool the proper work object by following the procedure in *Selecting tool, work object, and payload on page 112*. You do not need to perform this step if you previously selected the tool and/or work object.
4.	Tap **Coordinate System**, then tap **Tool** followed by **OK**.
5.	Press and hold the enabling device to activate the mechanical unit's motors.
6.	Move the joystick and the mechanical unit moves along.

Continues on next page

TIP!

Use the **QuickSet** menu to select jogging mode faster.

Jog with a stationary tool

If your robot system uses stationary tools, you must select both the proper tool and the proper work object (held by the robot) to jog in tool coordinates.

The tool coordinate system is defined by the position and orientation of the stationary tool and is fixed in space. To perform the intended operations you move the work object. This way positions can be expressed in the tool coordinate system.

5.3.10. Locking the joystick in specific directions

Overview

The joystick can be locked in specific directions to prevent movement for one or more axes.

This may be useful for instance while fine tuning positions or when programming operations that should only be performed in the direction of a specific coordinate system axis.

Note that the axes locked depends on the currently selected motion mode.

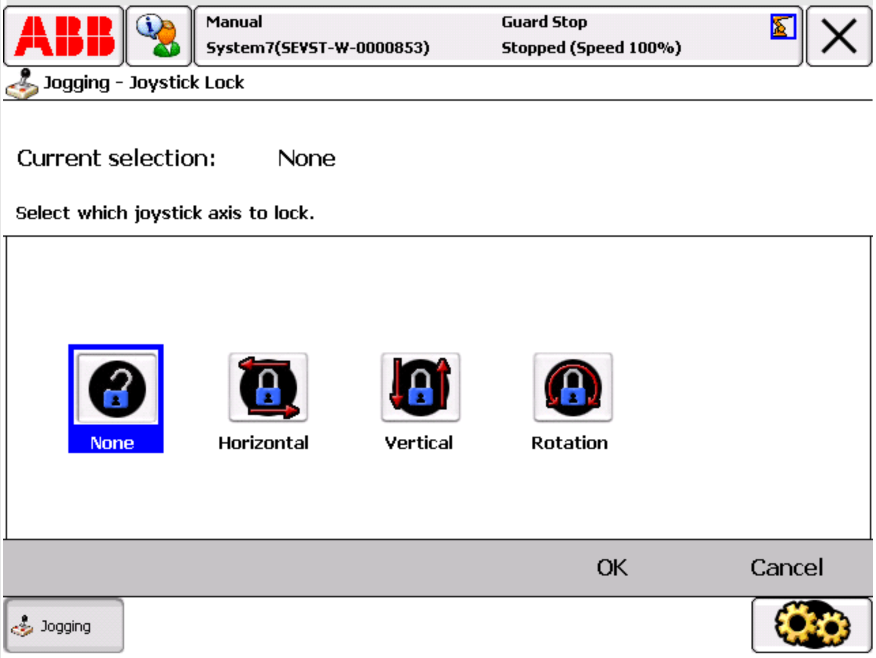
Which axes are locked?

This section describes how to see which joystick directions are locked

Step	Action
1.	On the **ABB** menu, tap **Jogging** to view jogging properties.
2.	Tap **Joystick lock** to check the joystick properties, or check the **Joystick directions** area properties in the right hand corner of the window. A padlock symbol is displayed for locked axes.

Locking the joystick in specific directions

This section describes how to lock the joystick in specific directions.

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Joystick lock**.  en0300000488
3.	Tap the joystick axis or axes that should be locked. The axis toggles between locked and unlocked each time you tap.
4.	Tap **OK** to lock.

Continues on next page

Unlocking all axes

This section describes how to unlock all axes from the joystick directions lock.

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Joystick lock**.
3.	Tap **None**, then tap **OK**.

5.3.11. Incremental movement for precise positioning

Incremental movement

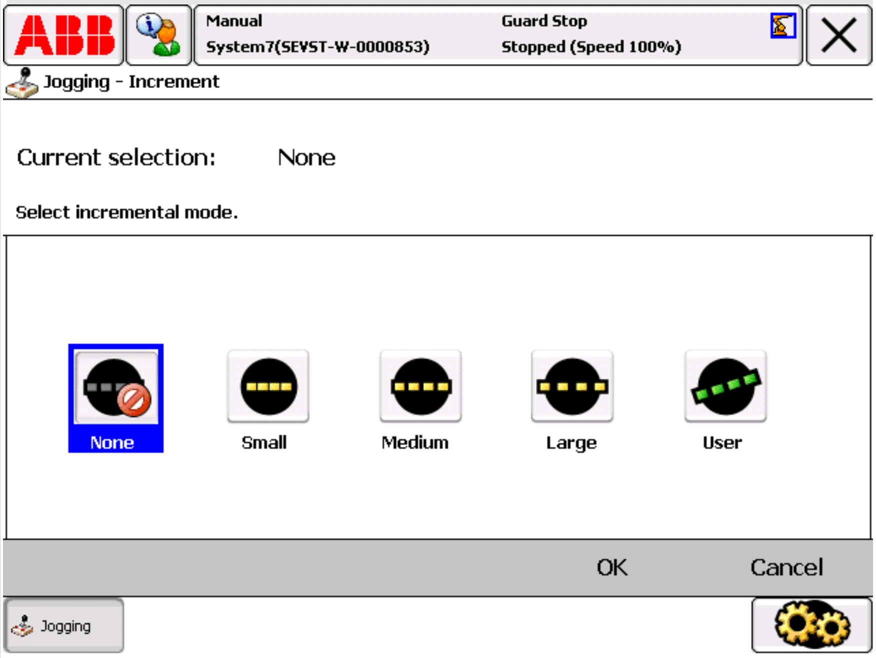
Use incremental movement to jog the robot in small steps, which enables very precise positioning.

This means that each time the joystick is deflected, the robot moves one step (increment). If the joystick is deflected for one or more seconds, a sequence of steps, (at a rate of 10 steps per second), will be performed as long as the joystick is deflected.

Default mode is no increment, then the robot moves continuously when the joystick is deflected.

Set the incremental movement size

This procedure details how to specify the size of the incremental movement.

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Increment**.  en0400000971
3.	Tap the desired increment mode, see description in section *Incremental movement sizes on page 125*.
4.	Tap **OK**.

Continues on next page

Incremental movement sizes

Choose between small, medium or large increments. You can also define your own increment movement sizes.

Increment	Distance	Angular
Small	0.05 mm	0.005°
Medium	1 mm	0.02°
Large	5 mm	0.2°
User		

5.3.12. How to read the exact position

How robot positions are displayed

Positions are always displayed as:

- the point in space expressed in the x, y and z tool center point coordinates,
- the angular rotation of the tool center point expressed in Euler angles or as a quaternion.

How additional axes' positions are displayed

When an additional axis is moved, only the axis position is displayed.

Linear axis positions are displayed in millimeters expressed as the distance to the calibration position.

Rotating axis positions are displayed in degrees expressed as the angle to the calibration position.

No positions displayed

No position is displayed when the mechanical unit is uncalibrated. Instead the text “Unit not calibrated" is displayed.

How to read the exact position

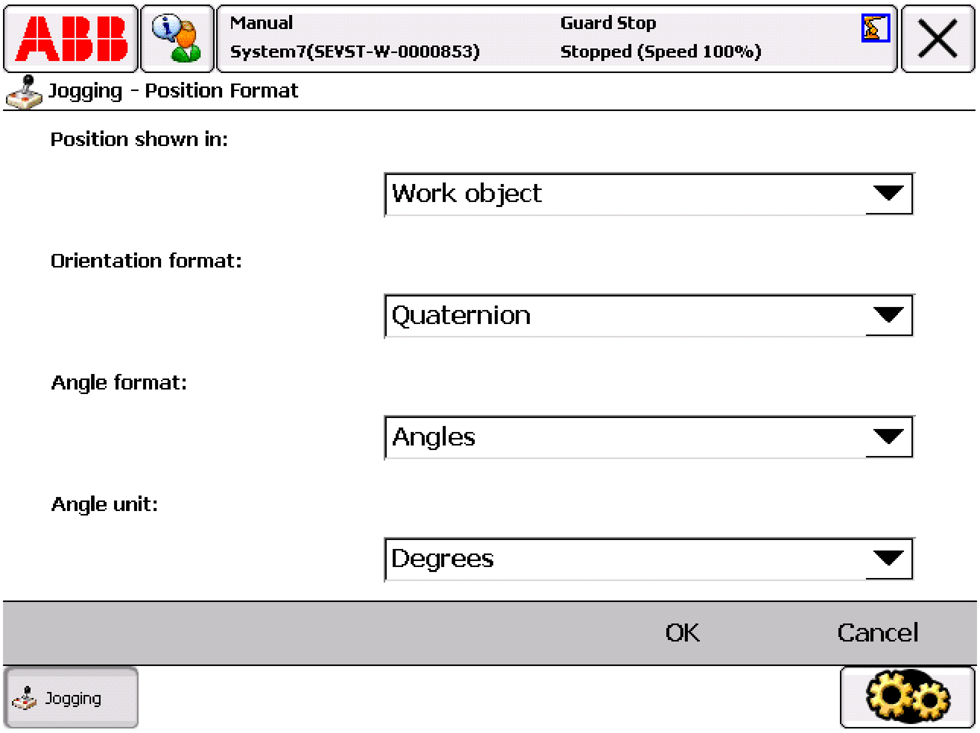
This procedure describes how to read the exact position.

Step	Action
1.	On the **ABB** menu tap **Jogging**.
2.	The position is displayed in the **Position** area properties in the right hand side of the window. See illustration in *Jogging on page 77*.

Illustration of position format

The illustration shows all available settings for showing the position format:

Continues on next page



en0400000991

Available selections for position format

The following selections are available for *Position shown* in *Illustration of position format on page 126*:

- World
- Base
- Work object

The following selections are available for *Orientation format*:

- Quaternion
- Euler angles

The following selections are available for *Position angle format*:

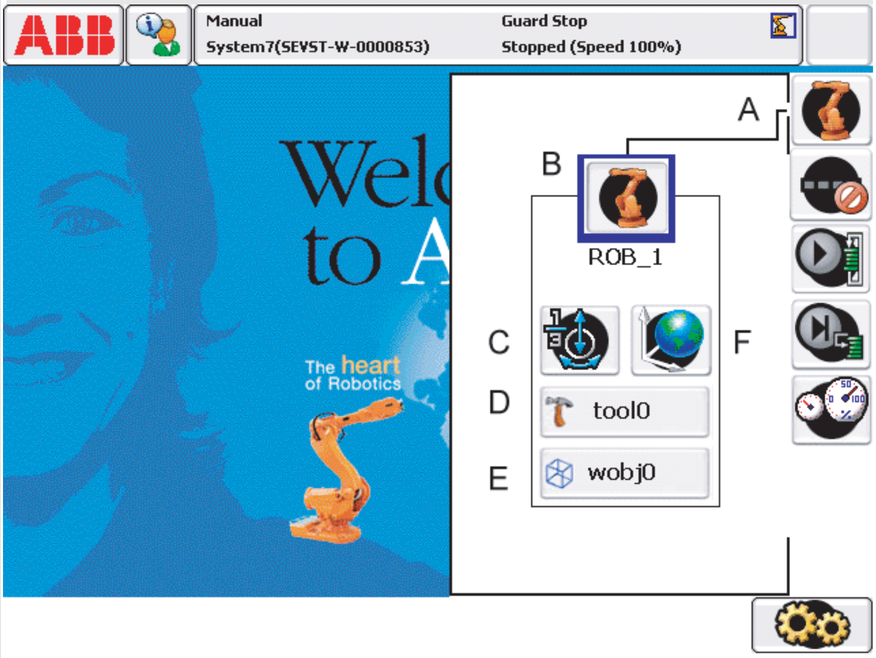
- Angles

The following selections are available for *Presentation angle unit*:

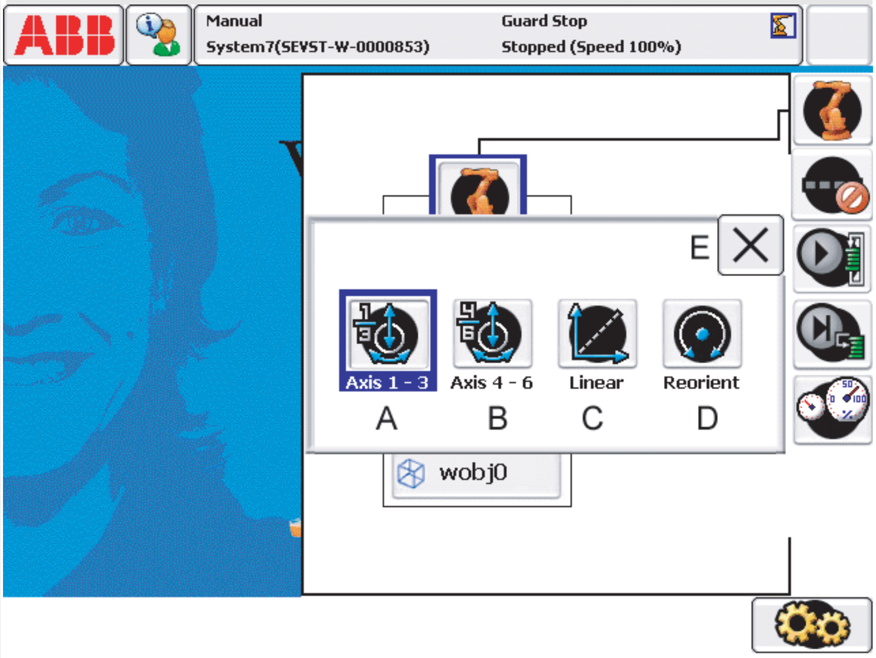
- Degrees
- Radians

5.3.13. Quickset menu, Mechanical unit

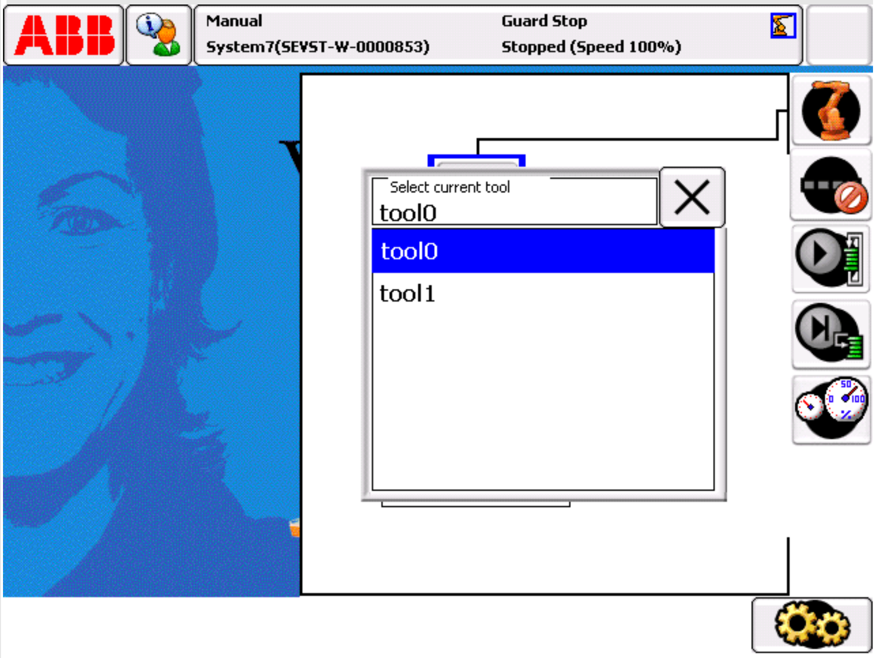
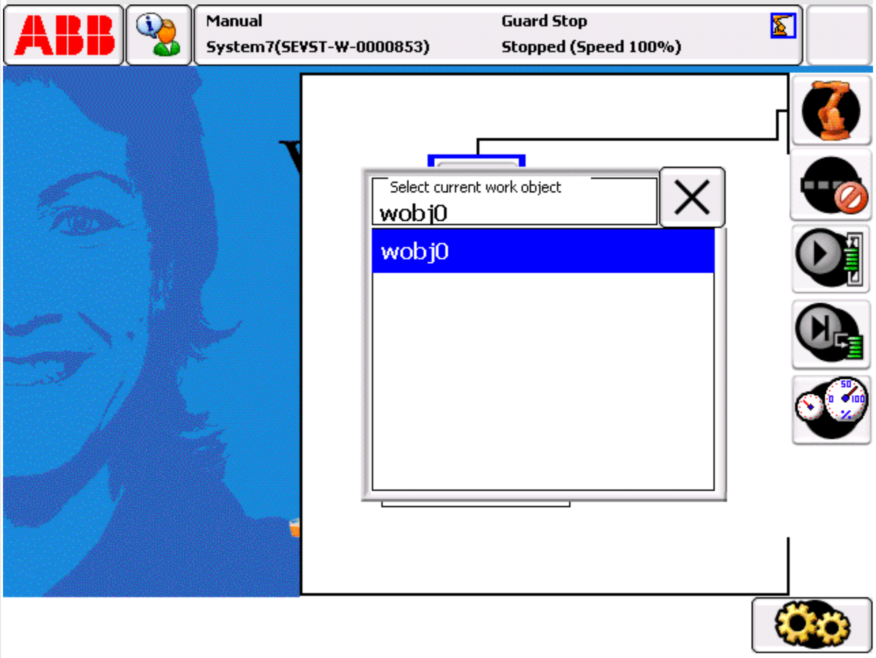
Mechanical unit button

Step	Action
1.	On the **Quickset** menu, tap Mechanical unit, then tap to select a mechanical unit.  en0300000539 The following buttons are displayed: • A: Mechanical unit menu button • B: Mechanical unit, a selected unit is highlighted • C: Motion mode settings • D: Tool settings • E: Work object settings • F: Coordinate system settings Each button is described in the following steps.

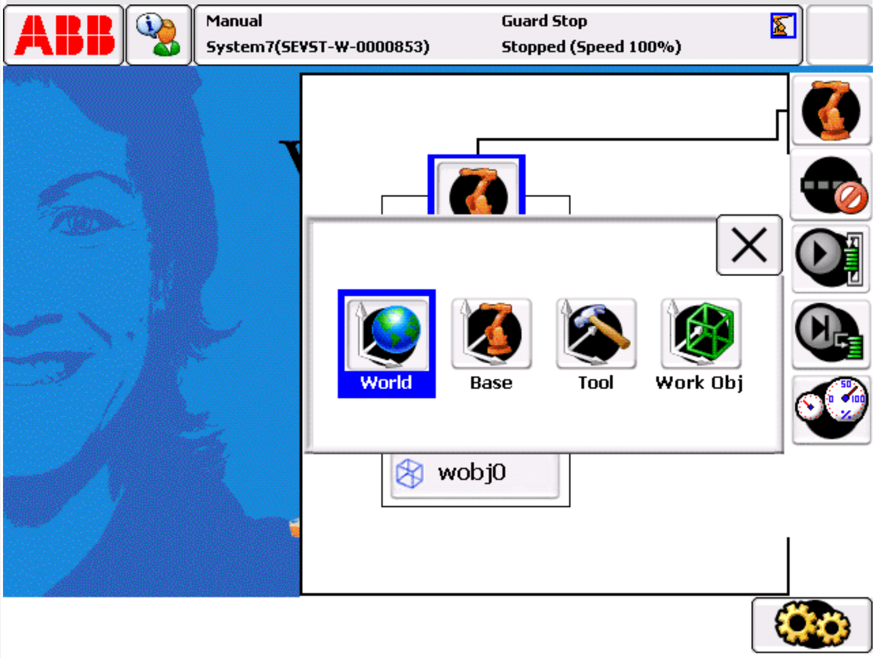
Continues on next page

Step	Action
2.	If you want to view/change any motion mode functionality, tap the Motion mode settings button.  en0300000540 The following buttons are displayed: • A: Axes 1-3 motion mode • B: Axes 4-6 motion mode • C: Linear motion mode • D: Reorient motion mode • E: Close motion mode settings

Continues on next page

Step	Action
3.	If you want to view/change the available tools, tap the Tool settings button.  en0400000988 A list of all available, defined tools is shown. Tap the one to use.
4.	If you want to view/change the available work objects, tap the Work object settings button.  en0400000989 A list of all available, defined work objects is shown. Tap the one to use.

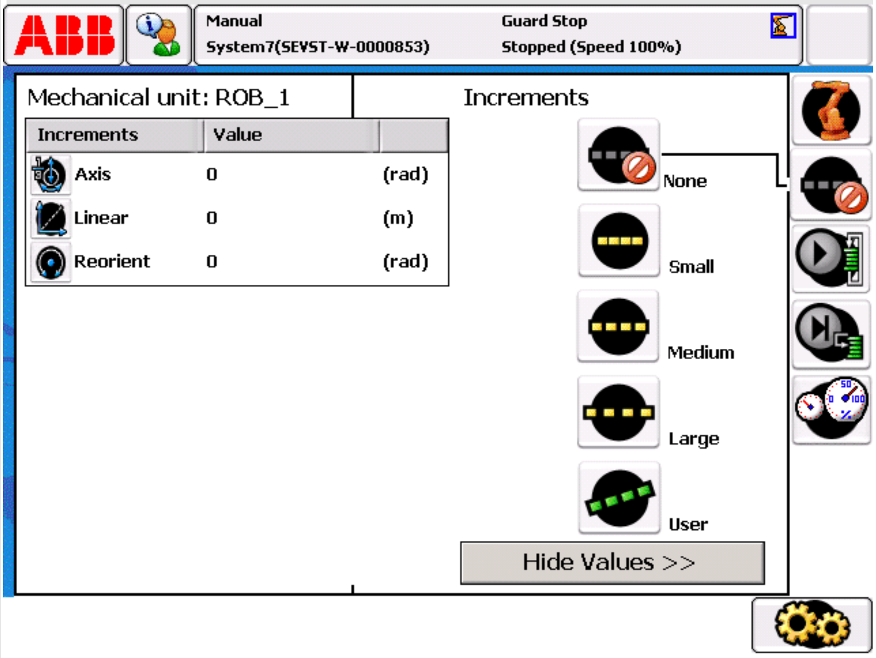
Continues on next page

Step	Action
5.	If you want to view/change any Coordinate system functionality, tap the Coordinate system settings button.  en0300000541 The following buttons are displayed: • World coordinate system • Base coordinate system • Tool coordinate system • Work object coordinate system

5.3.14. Quickset menu, Increment

Increment

All functions under this button may also be reached from the Jogging menu.

Step	Action
1.	If you want to view/change any increment functionality, tap the increment button.  en0300000542 Tap to select increment size: • None for no increments • Small movement increments • Medium movement increments • Large movement increments • User for movement increments defined by the user. Tap **Show values** to display the increment values.

6 Programming and testing

6.1. Before you start programming

Programming tools

You can use both the FlexPendant and RobotStudio$^{\text{Online}}$ for programming. For basic programming it may be easier to use RobotStudio$^{\text{Online}}$, while the FlexPendant is best suited for modifying programs, such as positions and paths.

How to program using RobotStudio$^{\text{Online}}$ is described in *Operator's manual - RobotStudio Online*.

Define tools, payloads and work objects

Define tools, payloads and work objects before you start programming. You can always go back and define more objects later, but you should define your basic objects in advance.

Define coordinate systems

Make sure the base and world coordinate systems have been set up properly during the installation of your robot system. Also make sure that additional axes have been set up.

Define tool and work object coordinate systems as needed before you start programming. As you add more objects later you also need to define the corresponding coordinate systems.

TIP!

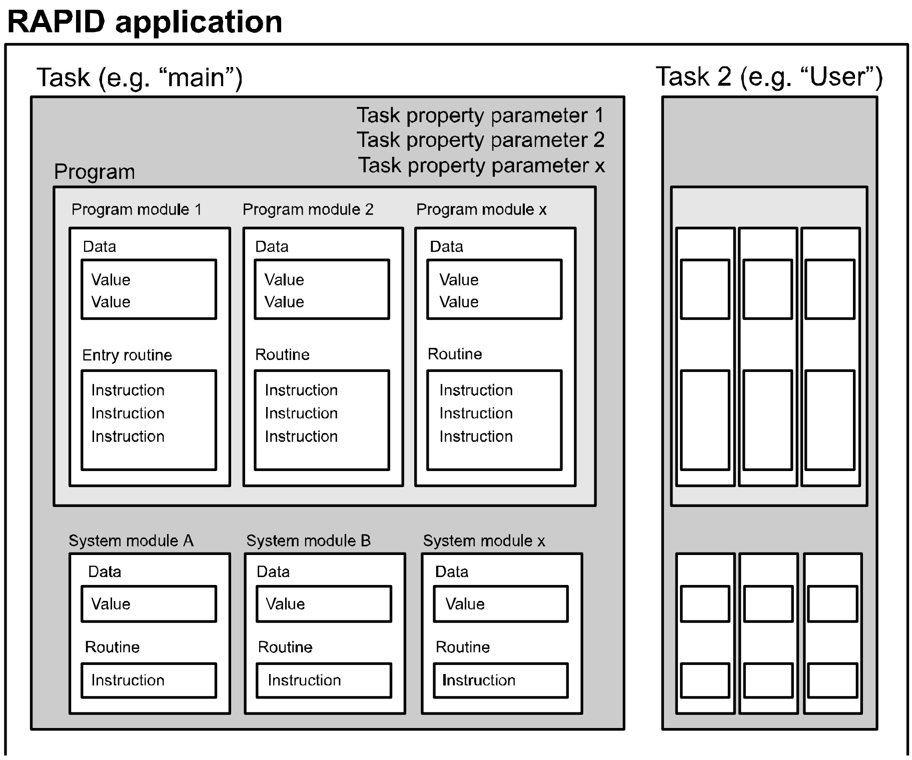
Need to know more about the RAPID language and structure? See the manuals *RAPID reference manual - RAPID overview*, *RAPID reference manual - Instructions*, and *RAPID reference manual - Functions and data types*.

6.2 Programming concept

6.2.1. The structure of a RAPID application

Illustration of a RAPID application

The structure and contents of the main computer RAM memory during operation is described in section *The structure of the main computer RAM memory contents on page 249*



en0300000576

Parts

Part	Function
Task	Each task usually contains a RAPID program and system modules aimed at performing a certain function, e.g. spot welding or manipulator movements. A RAPID application may contain one task. If you have the Multitasking option installed, then there can be more than one task. Read more about Multitasking in *Application manual - Engineering tools*.
Task property parameter	The task property parameters set certain properties for all task contents. Any program stored in a certain task, assumes the properties set for that task. The task property parameters are specified in *RAPID reference manual*.
Program	Each program usually contains program modules with RAPID code for different purposes. Any program must have an entry routine defined to be executable.

Continues on next page

Part	Function
Program module	Each program module contains data and routines for a certain purpose. The program is divided into modules mainly to enhance overview and facilitate handling the program. Each module typically represents one particular robot action or similar. All program modules will be removed when deleting a program from the controller program memory. Program modules are usually written by the user.
Data	Data are values and definitions set in program or system modules. The data are referenced by the instructions in the same module or in a number of modules (availability depending on data type). Data type definitions are specified in the *RAPID reference manual, Functions and data types*.
Routine	A routine contains sets of instructions, i.e. defines what the robot system actually does. A routine may also contain data required for the instructions.
Entry routine	A special type of routine, in English sometimes referred to as "main", defined as the program execution starting point. Note Each program **must** have an entry routine called "main", or it will not be executable. How to appoint a routine as entry routine is specified in *RAPID reference manual*. The default name for main can be changed by the system parameter configurations, type *Task*. See *Technical reference manual - System parameters*.
Instruction	Each instruction is a request for a certain event to take place, e.g. "Run the manipulator TCP to a certain position" or "Set a specific digital output". The instructions, their syntax and function is thoroughly described in the *RAPID reference manual, Instructions*.
System module	Each system module contains data and routines to perform a certain function. The program is divided into modules mainly to enhance overview and facilitate handling the program. Each module typically represents one particular robot action or similar. All system modules will be retained when "Delete program" is ordered. System modules are usually written by the robot manufacturer or line builder.

6.3 Data types

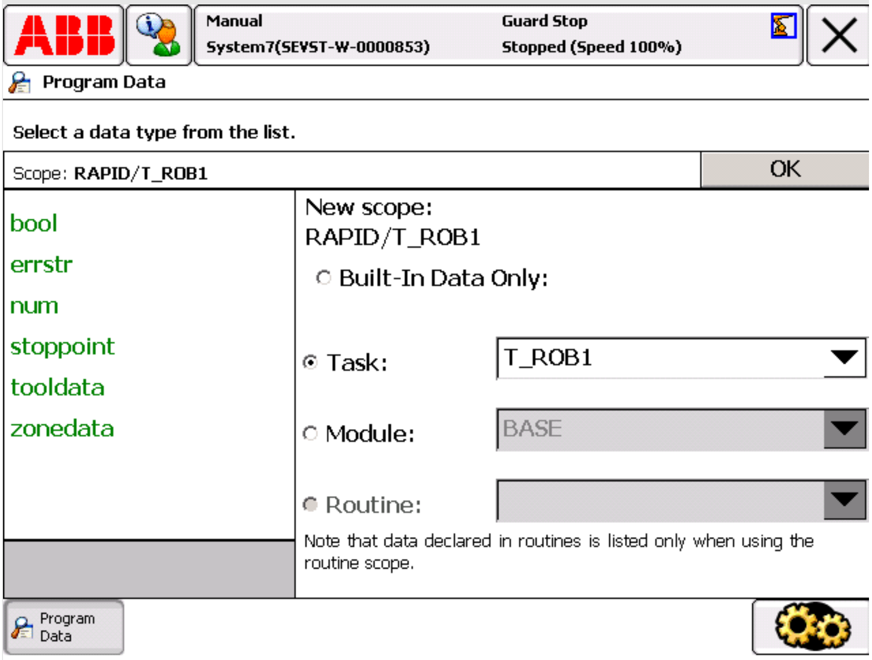
6.3.1. Viewing data in specific tasks, modules, or routines

Overview

It is possible to view selections of data types by selecting a specific scope.

Viewing data in specific tasks, modules, or routines

This section details how to view data instances in specific modules or routines.

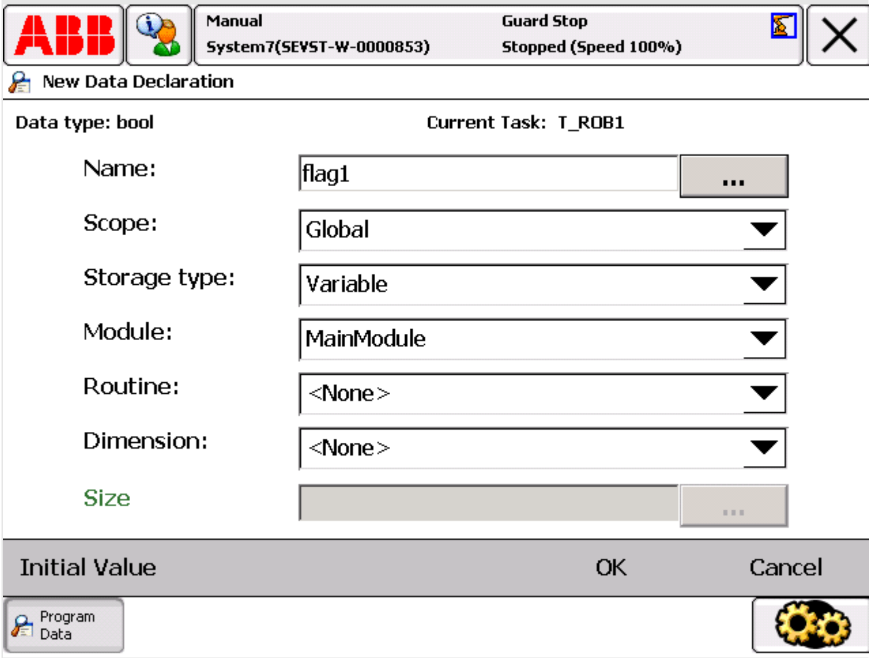
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>In the ABB menu, tap Program Data.</td></tr>
<tr><td>2.</td><td>Tap Change Scope.
The following screen is displayed:


en0400000661</td></tr>
<tr><td>3.</td><td>Select the required scope by selecting:
Built-In Data Only: Shows all data types used by the specific system
Task: Shows all data types used by a specific task
Module: Shows all data types used by a specific module
Routine: Shows all data types used by a specific routine</td></tr>
<tr><td>4.</td><td>Tap OK to confirm your choice.</td></tr>
<tr><td>5.</td><td>Tap twice to select a data type and view its instances.</td></tr>
</table>

6.3.2. Creating new data instance

Creating new data instance

This section details how to create new data instances of data types.

Step	Action
1.	In the **ABB** menu, tap **Program Data**. A list of all available data types is displayed.
2.	Tap the data instance type to be created, i.e. **bool** and then tap **Show data**. A list of all instances of the data type is displayed.
3.	Tap **New**.  en0400000663
4.	Tap **...** the right of **Name** to define the data instance's name.
5.	Tap the **Scope** menu to set accessibility for the data instance. Select: • **Global** • **Local** • **Task**
6.	Tap the **Storage type** menu to select type of memory used for the data instance. Select: • **Persistent** if the data instance is persistent • **Variable** if the data instance is variable • **Constant** if the data instance is constant
7.	Tap the **Module** menu to select module.
8.	Tap the **Routine** menu to select routine.

Continues on next page

Step	Action
9.	If you want to create an array of data instances, then tap the **Dimensions** menu and select the number of dimensions in the array, 1-3. • 1 • 2 • 3 • None Then tap **...** to set the **Size** of the array's axes. Arrays are described in section *What is a data array? on page 312*
10.	Tap **OK**.

6.3.3. Editing data instances

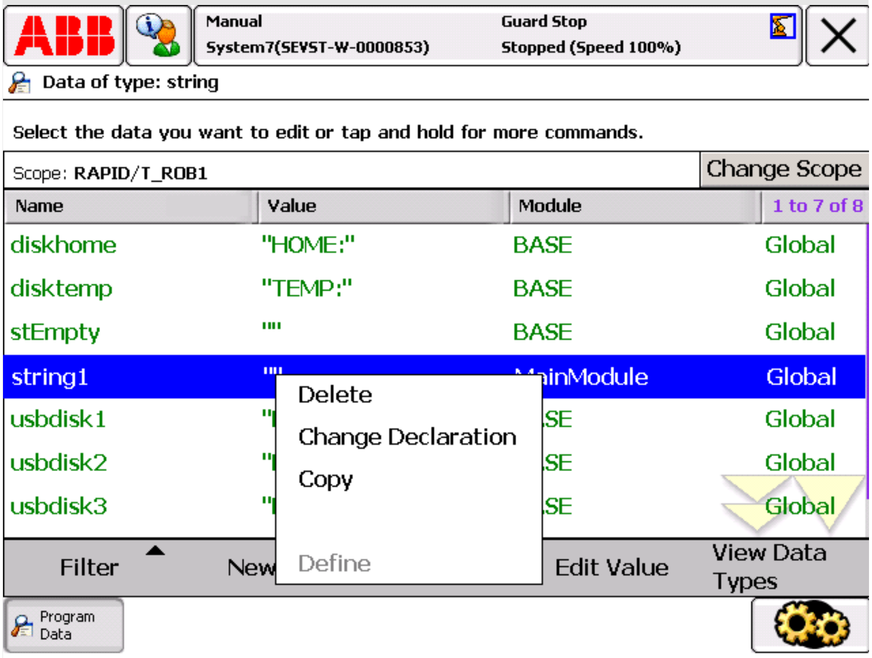
Overview

This section describes how to view data instances and then edit values, delete, change declaration of, copy, or define an instance.

For instances of the types tooldata, wobjdata, and loaddata, see sections *Tools on page 142*, *Work objects on page 154*, or *Payloads on page 161*.

Viewing data instances

This section details how to view the available instances of a data type.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>In the ABB menu, tap Program Data.</td></tr>
<tr><td>2.</td><td>Tap the data type of the instance you want view, and then tap Show Data.</td></tr>
<tr><td>3.</td><td>Tap and hold the data instance for a couple of seconds.
A circle of red dots and a menu appears.


en0400000671</td></tr>
<tr><td>4.</td><td>Depending on what you want to do with the data instance, you have the following choices:

Tap Delete to remove the data instance.
Tap Change Declaration to change the declaration of the data instance.
Tap Copy to copy the data instance.
Tap Define to define the tool frame (only available for tool, work object, and load data).

Proceed as described in the following sections.</td></tr>
</table>

Continues on next page

Editing data instance values

This section describes how to edit a data instance value..

Step	Action	Info
1.	Tap **Edit value** to open the instance.	
2.	Tap the value to open a keyboard or list of choices.	The way to edit values depend on the data type and possible values, for instance text, numbers, predefined values etc.
3.	Select or enter a new value.	
4.	Tap **OK**.	

Deleting data instances

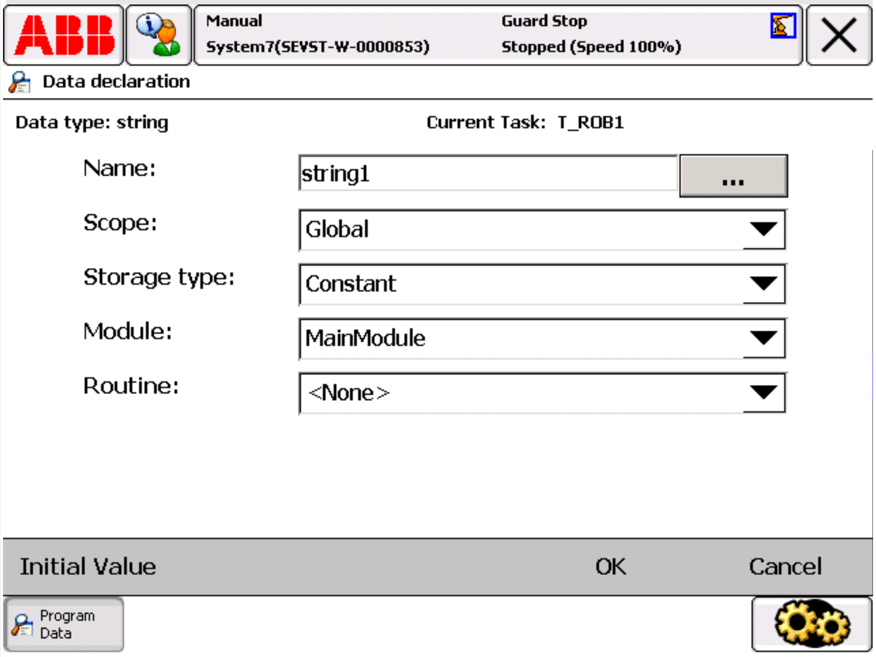
This section details how to delete data instances.

Step	Action
1.	Tap **Delete** in the menu for the data instance to be deleted, as detailed in section *Viewing data instances on page 139*. A dialog box is displayed.
2.	Tap **Yes** if you are sure the data instance is to be deleted.

Continues on next page

Changing data instance declarations

This section details how to change the declaration of the data instances.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>Tap Change declaration in the menu for the data instance to be deleted, as detailed in section Viewing data instances on page 139.


en0400000672</td></tr>
<tr><td>2.</td><td>Select what data instance values to be changed:

Name: Tap ... to bring out the soft keyboard and change the name.
Scope
Storage type
Module
Routine
</td></tr>
</table>

Copying data instances

This section details how to copy the data instances.

Step	Action
1.	Tap **Copy** in the menu for the data instance to be copied, as detailed in section *Viewing data instances on page 139*. A copy of the data instance is created. The copy has the same values as the original, but the name is unique.

Defining data instances

How to define tool frame or work object frame is described in sections *Defining the tool frame on page 145* or *Defining the work object coordinate system on page 155*.

6.4 Tools

6.4.1. Creating a tool

What happens when I create a tool?

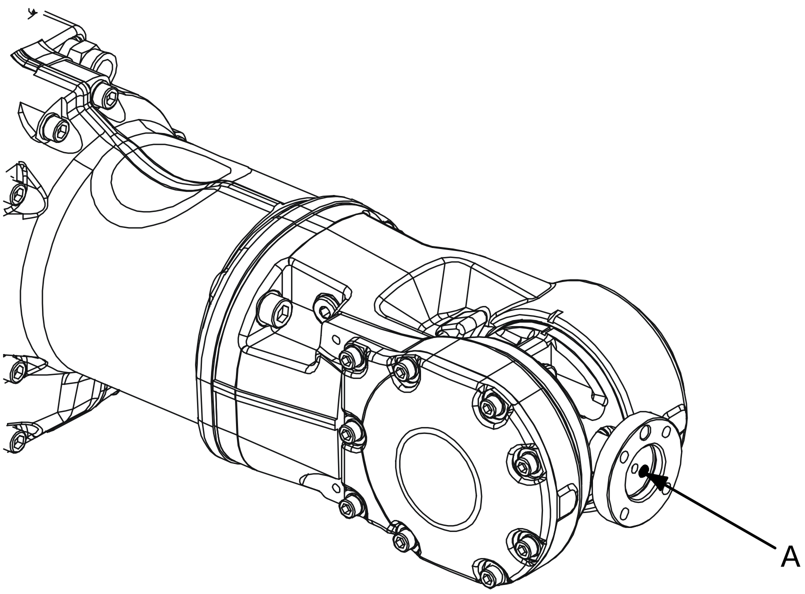
A variable of the type `tooldata` is created. The variable's name will be the name of the tool. For more information on data types, see *RAPID reference manual - Functions and data types*.

The new tool has default values for weight etc which must be defined before the tool can be used.

Creating a tool

The tool center point of the default tool (tool0) is in the center of the robot's mounting flange and with the mounting flange's orientation.

By creating a tool you can define another tool center point.

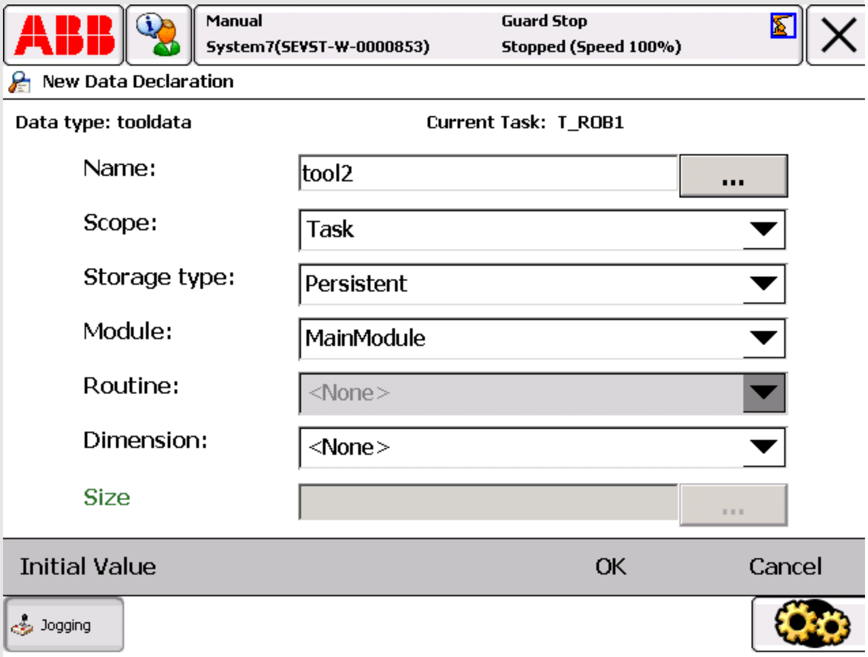


en0400000779

A	Tool center point, TCP, for tool0

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Tool** to display the list of available tools.

Continues on next page

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3.</td><td>Tap New... to create a new tool.


en0300000544
Enter values for each field, see table below.</td></tr>
<tr><td>4.</td><td>Tap OK.</td></tr>
</table>

Tool declaration settings

If you want to change...	then...	Recommendation
the tool's name	tap the ... button next to the name	Tools are automatically named `tool` followed by a running number, for example `tool10` or `tool21`. You should change this to something more descriptive such as gun, gripper or welder. If you change the name of a tool after it is referenced in any program you must also change all occurrences of that tool.
the scope	select the scope of choice from the menu	Tools should always be global to be available to all modules in the program.
the storage type	-	Tool variables must always be persistent.
the module	select the module in which this tool should be declared from the menu	

Continues on next page

NOTE!

The created tool is not useful until you have defined the tool data (TCP coordinates, weight etc.). This can be done by running the service routine LoadIdentify, or by editing the values manually. See *LoadIdentify, load identifiction service routine on page 214* or *Editing the tool data on page 148*.

6.4.2. Defining the tool frame

Preparations

To define the tool frame, you first need a reference point in the world coordinate system. If you need to set the tool center point orientation, you also need to affix elongators to the tool.

Selecting method for defining tool frame

This procedure describes how to select method for defining the tool frame.

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Tool** to display the list of available tools.
3.	Tap and hold the tool you want to define. A menu appears.
4.	In the menu, tap **Define...** The tool coordinate system definition dialog box appears.
5.	Select the method of choice from the **Method** pop up menu.
6.	Select the number of approach points from the **No of points** pop up menu.

Available methods

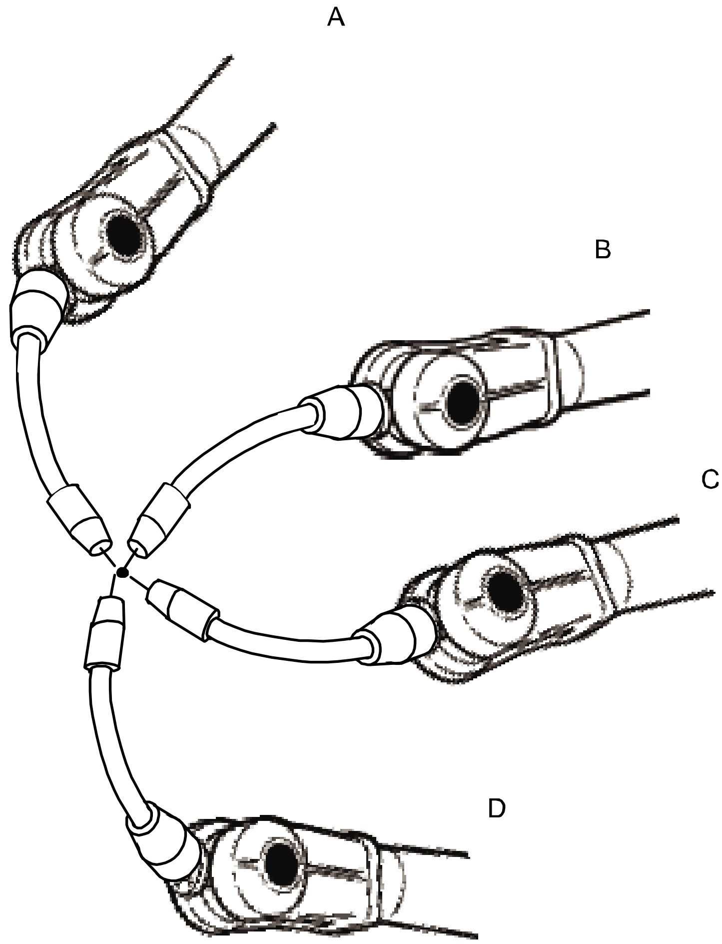
All methods require you to define the cartesian coordinates of the tool center point. The methods give you a choice in how the orientation will be set and defined.

If you want to...	...then select
set the orientation the same as the orientation of the robots mounting plate	4 points TCP
keep the current orientation	4 points TCP (orientation unchanged)
set the orientation in Z axis	5 points TCP & Z
set the orientation in X and Z axes	6 points TCP & Z, X

Continues on next page

Defining tool center point Cartesian coordinates

This procedure describes how to define the tool center point in Cartesian coordinates.



en0400000906

Step	Action	Info
1.	Jog the robot to an appropriate position for the first approach point.	Position A
2.	Tap **Modify Position** to define the point.	
3.	Repeat step 1 and 2 for each approach point to be defined, positions B, C, and D.	Jog away from the fixed world point to achieve best results. Just changing the tool orientation will not give as good a result.

Continues on next page

Defining elongator points

This procedure describes how to define the elongator points.

Step	Action
1.	Without changing the tool’s orientation, jog the robot so that the fixed world point becomes a point on the desired positive axis of the rotated tool coordinate system.
2.	Tap **Modify Position** to define the point.
3.	Repeat step 1 and 2 for the second axis if it also should be defined.

6.4.3. Editing the tool data

Tool data

Use the value settings to set the tool center point position and physical properties of the tool such as weight and center of gravity.

This can also be done automatically with the service routine LoadIdentify. See sections *Running a service routine on page 209*, or *LoadIdentify, load identifiction service routine on page 214*.

Displaying the tool data

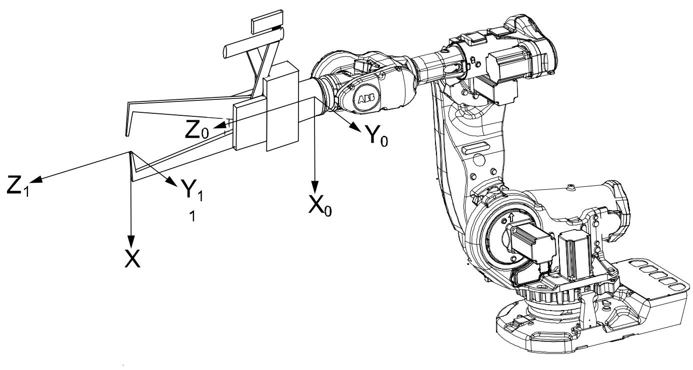
This section details how to display the tool data.

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Tool** to display the list of available tools.
3.	Tap and hold the tool you want to edit. A menu appears. • Change Declaration • Change Value • Delete • Define
4.	In the menu, tap **Change Value**. The data that defines the tool appears. Green text indicates that the value can be changed.
5.	Proceed with changing the data as described below.

Measuring the tool center point

The easiest way to define the tool center point, TCP, is usually to use the predefined method described in *Defining the tool frame on page 145*. If you use this method, you do not have to write any values for tframe as these are supplied by the method.

If you already have the measurements of the tool, or for some reason want to measure them manually, the values can be entered in the tool data.



en0400000881

Continues on next page

X_0	X axis for tool0
Y_0	Y axis for tool0
Z_0	Z axis for tool0
X_1	X axis for the tool you want to define
Y_1	Y axis for the tool you want to define
Z_1	Z axis for the tool you want to define

Step	Action
1.	Measure the distance from the center of the robot's mounting flange to the tool's center point along the X axis of tool0.
2.	Measure the distance from the center of the robot's mounting flange to the tool's center point along the Y axis of tool0.
3.	Measure the distance from the center of the robot's mounting flange to the tool's center point along the Z axis of tool0.

Editing the tool definition

Step	Action	Instance	Unit

1.	Enter the cartesian coordinates of the tool center point's position.	`tframe.trans.x`
		`tframe.trans.y`
		`tframe.trans.z`
			[mm]

2.	If necessary, enter the tool frame orientation.	`tframe.rot.q1`
		`tframe.rot.q2`
		`tframe.rot.q3`
		`tframe.rot.q4`
			None

3.	Enter the weight of the tool.	`tload.mass`	[kg]

4.	If necessary, enter the tool's center of gravity.	`tload.cog.x`
		`tload.cog.y`
		`tload.cog.z`
			[mm]

5.	If necessary, enter the orientation of the axis of moment	`tload.aom.q1`
		`tload.aom.q2`
		`tload.aom.q3`
		`tload.aom.q4`
			None

6.	If necessary, enter the tool's moment of inertia.	`tload.ix`
		`tload.iy`
		`tload.iz`
			[kgm^2]

7.	Tap **OK** to use the new values, **Cancel** to leave the definition unchanged.		

6.4.4. Editing the tool declaration

Tool declaration

Use the declaration to change how the tool variable can be used in the program’s modules.

Displaying the tool declaration

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Tool** to see the list of available tools.
3.	Tap and hold the tool whose definition you want to edit. A menu appears. • Change Declaration • Change Value • Delete • Define
4.	In the menu, tap **Change Declaration**. The tool's declaration appears.
5.	Edit the tool declaration as listed in section *Creating a tool on page 142*.

NOTE!

If you change the name of a tool after it is referenced in any program you must also change all occurrences of that tool.

6.4.5. Deleting a tool

Deleting a tool

This section describes how to delete a tool.

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Tool** to display the list of available tools.
3.	Tap and hold the tool you want to delete. A menu appears.
4.	Tap **Delete** to delete the selected tool. A confirmation dialog box appears.
5.	In the dialog box, tap **Yes** to delete the tool, **No** to keep the tool.

CAUTION!

A deleted tool, work object or payload cannot be recovered, and all related data will be lost. If the tool, work object or payload is referenced by any program, those programs cannot run without changes.

If you delete a tool you cannot continue the program from the current position.

6.4.6. Setup for stationary tools

Stationary tools

Stationary tools are used, for instance, in applications that involve large machines such as cutters, presses and punch cutters. You may use stationary tools to perform any operation that would be difficult or inconvenient to perform with the tool on the robot.

With stationary tools, the robot holds the work object.

Make a tool stationary

This section describes how to make a tool stationery.

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Tool** to display the list of available tools.
3.	Tap and hold the tool you want to edit. A menu appears.
4.	In the menu, tap **Change value**. The data that defines the tool appears.
5.	Tap the instance `robhold`.
6.	Tap **FALSE** to make this tool stationary.
7.	Tap **OK** to use the new setup, **Cancel** to leave the tool unchanged.

Make a work object robot held

This section describes how to make a work object robot held.

Step	Action
1.	In the Jogging window, tap **Work object** to display the list of available work objects.
2.	Tap and hold the work object you want to edit. A menu appears.
3.	In the menu, tap **Change value**. The data that defines the work object appears.
4.	Tap the instance `robhold`.
5.	Tap **TRUE** to indicate that this work object is held by the robot.
6.	Tap **OK** to use the new setup, **Cancel** to leave the work object unchanged.

Continues on next page

Differences in coordinate system referencing

This section describes differences in coordinate system referencing.

The...	...normally references the...	...but now references the...
work object coordinate system	user coordinate system	user coordinate system (no change)
user coordinate system	world coordinate system	robot's mounting plate
tool coordinate system	robot's mounting plate	world coordinate system

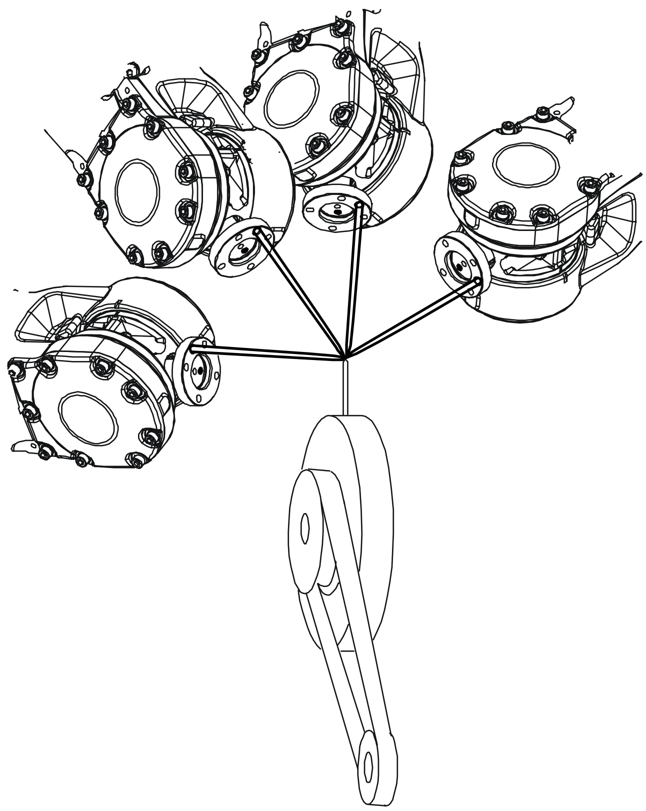
Set up the tool coordinate system

You use the same measurement methods to set up a stationary tool coordinate system as with tools mounted on the robot.

The world reference tip must, in this case, be attached to the robot. Define and use a tool with the reference tip's measurements when you create approach points. You also need to attach elongators to the stationary tool if you need to set up the orientation.

You should enter the reference tip's tool definition manually to minimize errors when calculating the stationary tool's coordinate system.

You may enter the stationary tool's definition manually.



en0400000990

6.5 Work objects

6.5.1. Creating a work object

What happens when I create a work object?

A variable of the type `wobjdata` is created. The variable's name will be the name of the work object.For more information on data types, see *RAPID reference manual - Functions and data types*.

See also *What is a work object? on page 304* for more details.

Creating a work object

The work object's coordinate system is now identical with the world coordinate system. To define the position and orientation of the work object's coordinate system, see Editing the work object data on page 158

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Work Object** to display the list of available work objects.
3.	Tap **New...** to create a new work object.
4.	Tap **OK**.

Work object declaration settings

If you want to change...	then...	Recommendation
the work object's name	tap the **...** button next to it	Work objects are automatically named `wobj` followed by a running number, for example `wobj10`, `wobj27`. You should change this to something more descriptive. If you change the name of a work object after it is referenced in any program you must also change all occurrences of that work object.
the scope	select the scope of choice from the menu	Work objects should always be global to be available to all modules in the program.
the storage type	-	Work object variables must always be persistent.
the module	select the module in which this work object should be declared from the menu	

6.5.2. Defining the work object coordinate system

Overview

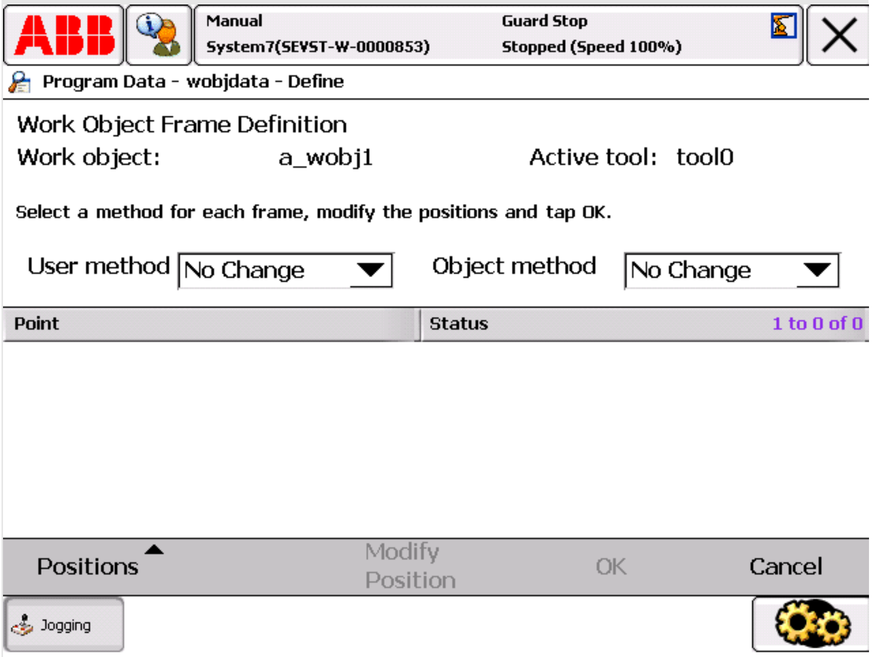
A work object must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).

When defining a work object you can use either the user frame or the object frame.

Selecting method

This procedure describes how to select method. Note that this only works for a user created work object, not the default work object, wobj0.

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Tap **Work object** to display the list of available work objects.
3.	Tap and hold the work object you want to define. A menu appears.
4.	In the menu, tap **Define...**.
5.	Select method from the **User method** or the **Object method** menu. See *Defining the user frame on page 156* and *Defining the work object frame on page 157*

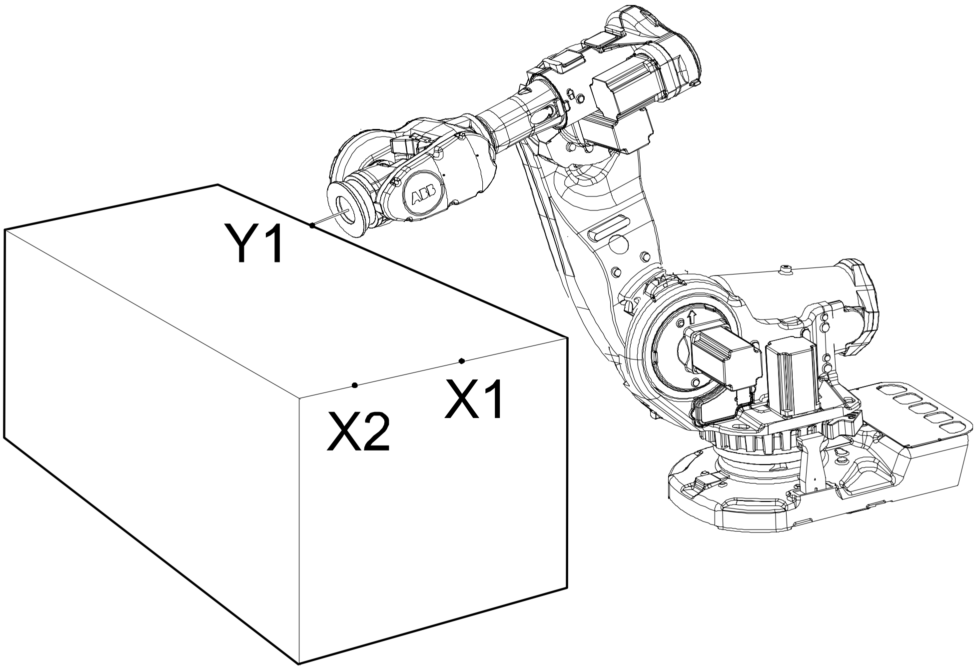


en0400000893

Continues on next page

Defining the user frame

This section details how to define the user frame. Note that this only works for a user created work object, not the default work object, wobj0.



en0400000887

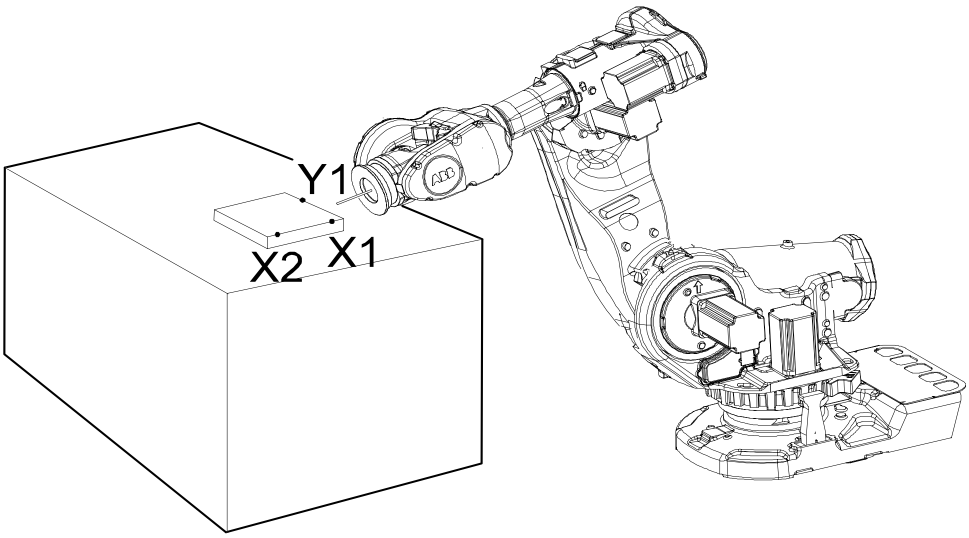
The x axis will go through points X1-X2, and the y axis through Y1.

Step	Action	Info
1.	In the **User method** pop up menu, tap **3 points**.	
2.	Press the enabling device and jog the robot to the first (X1, X2 or Y1) point you want to define.	Great distance between X1 and X2 is preferable for a more precise definition.
3.	Select the point in the list.	
4.	Tap **Modify Position** to define the point.	
5.	Repeat steps 2 to 4 for the remaining points.	

Continues on next page

Defining the work object frame

This section describes how to define the work object frame.



en0400000899

The x axis will go through points X1-X2, and the y axis through Y1.

Step	Action
1.	In the **Object method** pop up menu, tap **3 points**.
2.	See steps 2 to 4 in the description of *Defining the user frame on page 156*.

6.5.3. Editing the work object data

Overview

Use the definition to set the position of the user and work object coordinate system and physical properties of the work object, such as weight and center of gravity.

Displaying the work object data

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Work object** to display the list of available work objects.
3.	Tap and hold the work object you want to edit. A menu appears.
4.	In the menu, tap **Change Value**. The data that defines the work object appears.

Setting the work object and user coordinate systems position

The easiest way to set the work object and user coordinate systems position is to use the method described in *Defining the work object coordinate system on page 155*.

Step	Action	Instance	Unit

1.	Enter the cartesian coordinates of the position of the work object coordinate system.	`oframe.trans.x`
		`oframe.trans.y`
		`oframe.trans.z`
			[mm]

2.	If necessary, enter the coordinate system's orientation.	`oframe.rot.q1`
		`oframe.rot.q2`
		`oframe.rot.q3`
		`oframe.rot.q4`
			-

3.	Enter the cartesian coordinates of the position of the user coordinate system.	`uframe.trans.x`
		`uframe.trans.y`
		`uframe.trans.z`
			[mm]

4.	If necessary, enter the coordinate system's orientation.	`uframe.rot.q1`
		`uframe.rot.q2`
		`uframe.rot.q3`
		`uframe.rot.q4`
			-

5.	Tap **OK** to user the new values, **Cancel** to leave the definition unchanged.	-	-

6.5.4. Editing the work object declaration

Overview

Use the declaration to change how the work object variable can be used in the program's modules.

Displaying the work object declaration

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Work object** to see the list of available work objects.
3.	Tap and hold the work object who's definition you want to edit. A menu appears.
4.	In the menu, tap **Change Declaration**.
5.	The work object's declaration appears.
6.	Edit the tool declaration as listed in section *Creating a work object on page 154*.

NOTE!

If you change the name of a work object after it is referenced in any program you must also change all occurrences of that work object.

6.5.5. Deleting a work object

Deleting a work object

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Work object** to display the list of available work objects.
3.	Tap and hold the work object you want to delete. A menu appears.
4.	In the menu, tap **Delete** to delete the work object. A confirmation dialog box appears.
5.	In the dialog box, tap **Yes** to delete the work object, **No** to keep it.

CAUTION!

A deleted tool, work object or payload cannot be recovered, and all related data will be lost. If the tool, work object or payload is referenced by any program, those programs cannot run without changes.

If you delete a tool you cannot continue the program from the current position.

6.6 Payloads

6.6.1. Creating a payload

What happens when I create a payload?

A variable of the type `loaddata` is created. The variables name will be the name of the payload. For more information on data types, see *RAPID reference manual - Functions and data types*.

Adding a new payload and setting data declaration

The payloads coordinate system will be set to the position, including orientation, of the world coordinate system.

Step	Action
1	In the **ABB** menu tap **Jogging**.
2	Tap **Payload** to display the list of available payloads.
	Tap **New** to create a new payload. Enter data, see table below.
3	Tap **OK**.

Payload declaration settings

If you want to change...	...then...	Recommendation
the payload's name	tap the *...* button next to it	Payloads are automatically named `load` followed by a running number, for example `load10`, `load31`. You should change this to something more descriptive. If you change the name of a payload after it is referenced in any program you must also change all occurrences of that payload's name.
the scope	select the scope of choice from the menu	Payloads should always be global to be available to all modules in the program.
the storage type	-	Payload variables must always be persistent.
the module	select the module in which this payload should be declared from the menu	-

6.6.2. Editing the payload data

Overview

Use the payload data to set physical properties of the payload such as weight and center of gravity.

This can also be done automatically with the service routine LoadIdentify. See sections *Running a service routine on page 209*, or *LoadIdentify, load identifiction service routine on page 214*.

Displaying the payload definition

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Payload** to display the list of available payloads.
3.	Tap and hold the payload you want to edit. A menu appears.
4.	On the menu, tap **Change Value**. The data that defines the payload appears.

Changing the payload data

This procedure describes how to manually enter the payload data. This can also be done automatically by running the service routine *Load Identify*. How to run a service routine is described in section *Running a service routine on page 209*.

Step	Action	Instance	Unit
1.	Enter the weight of the payload.	`load.mass`	[kg]

2.	Enter the payload's center of gravity.	`load.cog.x`
		`load.cog.y`
		`load.cog.z`
			[mm]

3.	Enter the orientation of the axis of moment.	`load.aom.q1`
		`load.aom.q2`
		`load.aom.q3`
		`load.aom.q3`	

4.	Enter the payload's moment of inertia.	`ix`
		`iy`
		`iz`
			[kgm^2]

5.	Tap **OK** to use the new values, **Cancel** to leave the data unchanged.	-	-

6.6.3. Editing the payload declaration

Overview

Use the declaration to change how the payload variable can be used in the program's modules.

Displaing the payload declaration

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Payload** to see the list of available payloads.
3.	Tap and hold the payload whose definition you want to edit. A menu appears.
4.	In the menu, tap **Change declaration**.
5.	The payload's declaration appears. See *Creating a payload on page 161*.

NOTE!

If you change the name of a payload after it is referenced in any program you must also change all occurrences of that payload's name.

6.6.4. Deleting a payload

Deleting a payload

Step	Action
1.	In the **ABB** menu, tap **Jogging**.
2.	Tap **Payload** to display the list of available payloads.
3.	Tap and hold the payload you want to delete. A menu appears.
4.	Tap **Delete**. A confirmation dialog box appears.
5.	In the dialog box, tap **Yes** to delete the payload, **No** to keep the payload.

CAUTION!

A deleted tool, work object or payload cannot be recovered, and all related data will be lost. If the tool, work object or payload is referenced by any program, those programs cannot run without changes.

If you delete a tool you cannot continue the program from the current position.

6.7 Programming

6.7.1. Handling of programs

Overview

This section details how to perform normal handling of existing robot programs. It details how to:

- create a new program
- load an existing program
- save a program
- rename a program
- delete a program

Each task must contain *one* program, no more, no less.

How to create a new program *when no program is available* is detailed in section *Creating a new program on page 165*.

About program files

When saving a program to the controller hard disk, it is by default saved to the directory HOME in the system's folder. The program is saved as a folder, named as the program, containing the actual program file, of type pgf.

When loading a program you open the program folder and select the pgf file.

When renaming a program you rename the program folder and the program file.

When saving a loaded program which is already saved to the hard disk, you must not open the existing program folder. Instead, you should save the program folder again and overwrite the old version, or rename the program.

Creating a new program

This section describes how to create a new program.

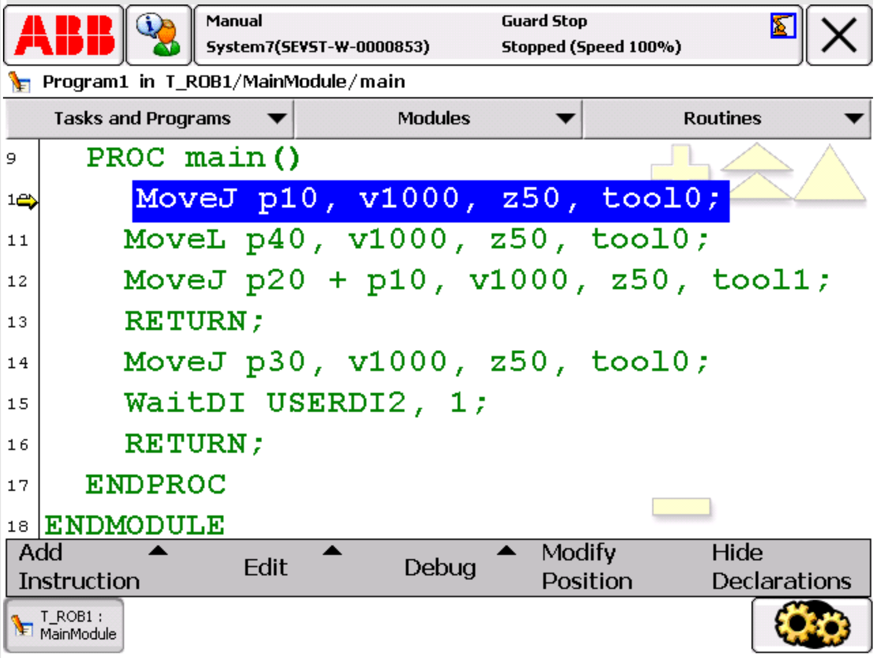
Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Tasks and Programs**.
3.	Tap **File**, then **New Program**. If there was already a program loaded, a warning dialog appears. • Tap **Save** to save the loaded program. • Tap **Don't save** to close loaded program without saving it, i.e. delete from program memory. • Tap **Cancel** to leave the program loaded.
4.	Use the soft keyboard to name the new program. Then tap **OK**.
5.	Tap **Back** to return to the editor view or **Show modules** to add existing modules to the new program. Continue by adding instructions, routines, or modules.

Continues on next page

Loading an existing program

This section describes how to load an existing program.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>On the ABB menu, tap Program Editor.</td></tr>
<tr><td>2.</td><td>Tap Tasks and Programs.</td></tr>
<tr><td>3.</td><td>Tap File, then Load Program.
If there was already a program loaded, a warning dialog appears.
Tap Save to save the loaded program.
Tap Don’t save to close loaded program without saving it, i.e. delete from program memory.
Tap Cancel to leave the program loaded.</td></tr>
<tr><td>4.</td><td>Use the file searching tool to locate the program file to be loaded (file type pgf). Then tap OK.
The program is loaded and the program code is displayed.



en0400000699</td></tr>
</table>

Saving a program

This section describes how to save a loaded program to the controller’s hard disk.

A loaded program is automatically saved in the program memory, but saving to the controller hard disk is an extra precaution.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Tasks and Programs**.

Continues on next page

Step	Action
3.	Tap **File** and select **Save Program As...**.
4.	Use the suggested program name or tap **...** to open the soft keyboard and enter a new name. Then tap **OK**.

Renaming a loaded program

This section describes how to rename a loaded program.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Tasks and Programs**.
3.	Tap **File** and select **Rename Program**. A soft keyboard is displayed.
4.	Use the soft keyboard to enter the new name of the program. Then tap **OK**.

6.7.2. Handling of modules

Overview

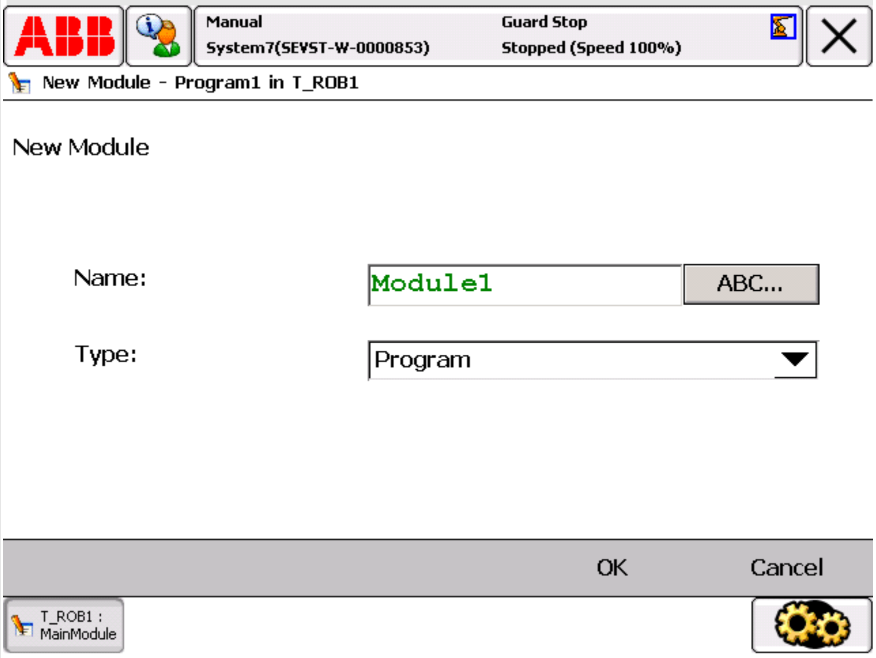
This section details how to handle program modules. i.e.:

- create a new module
- load an existing module
- save a module
- rename a module
- delete a module

Creating a new module

This section describes how to create a new module.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>On the ABB menu, tap Program Editor.</td></tr>
<tr><td>2.</td><td>Tap Modules.</td></tr>
<tr><td>3.</td><td>Tap File, then tap New Module.



en0400000688</td></tr>
<tr><td>4.</td><td>Tap ABC... and use the soft keyboard to enter the new module’s name. Then tap OK to close the soft keyboard.</td></tr>
<tr><td>5.</td><td>Select which type of module to be created:

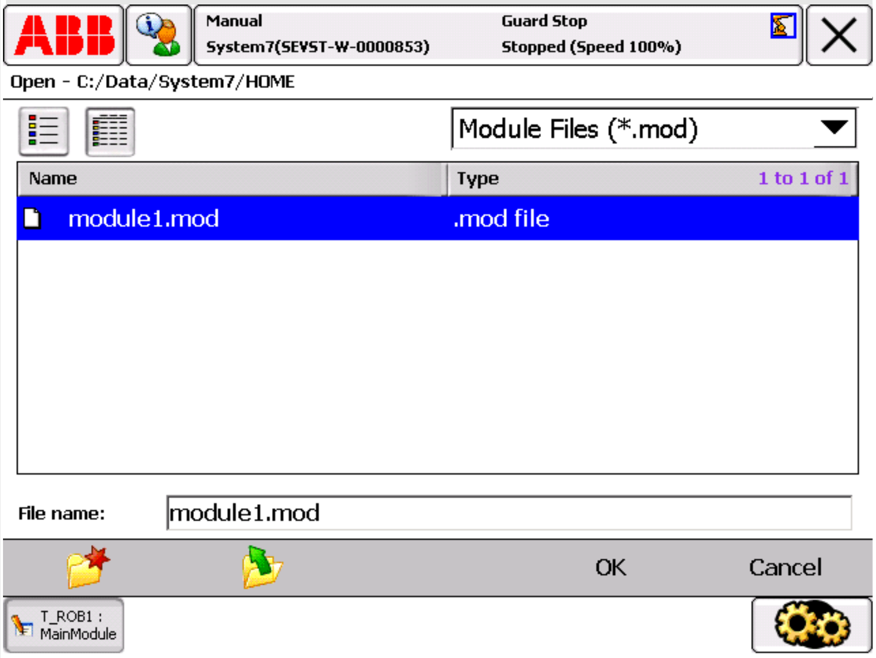
- Program
- System

Then tap OK.
The differences between module types are described in section The structure of a RAPID application on page 134.</td></tr>
</table>

Continues on next page

Loading an existing module

This section describes how to load an existing module.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Modules**.
3.	Tap **File**, then **Load Module**.  en0400000689 Locate the module to be loaded. See section *FlexPendant Explorer on page 75*.
4.	Tap **OK** to load the selected module. The module is loaded.

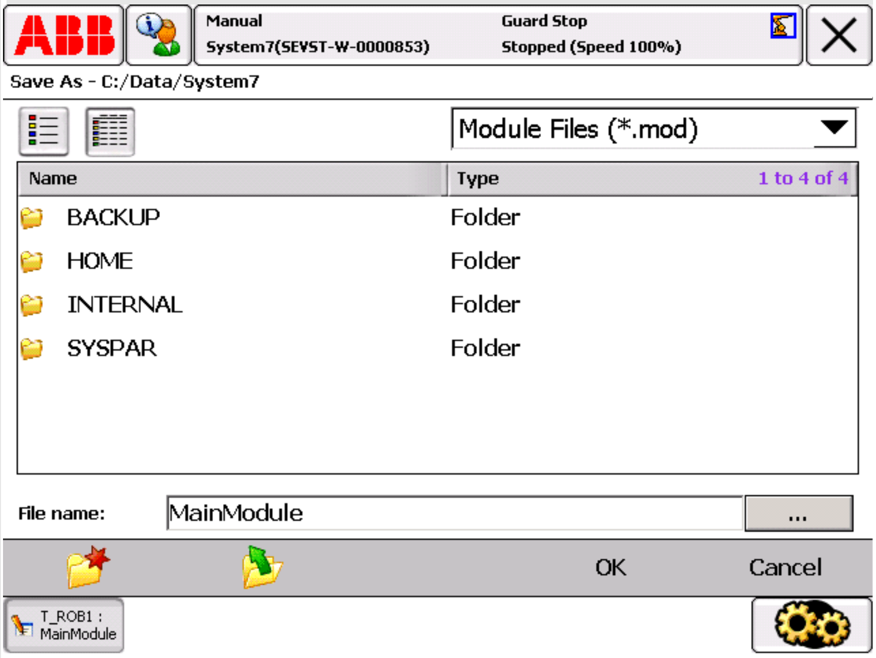
Saving a module

This section describes how to save a module.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Modules** and tap to select the module you want to load.

Continues on next page

Step	Action
3.	Tap **File**, then **Save Module As...**



en0400000690

Step	Action
4.	Tap on the suggested file name and use the soft keyboard to enter the module's name. Then tap **OK**.
5.	Use the file searching tool to locate where you want to save the module. See section *FlexPendant Explorer on page 75*. The default location is on the controller disk. The tap **OK**. The module is saved.

Renaming a module

This section describes how to rename a module.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Modules**.
3.	Tap **File**, then **Rename Module...** The soft keyboard is displayed.
4.	Use the soft keyboard to enter the module's name. Then tap **OK**.

Continues on next page

Deleting a module

This section describes how to delete a module from memory. If the module has been saved to disk, it will not be erased from the disk..

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Modules** and tap to select the module you want to delete.
3.	Tap **File**, then **Delete Module...** A dialog box is displayed.
4.	Tap **OK** to delete the module without saving it. If you want to save the module first, tap **Cancel** and save the module first. How to save the module is detailed in section *Saving a module on page 169*.

6.7.3. Handling of routines

Overview

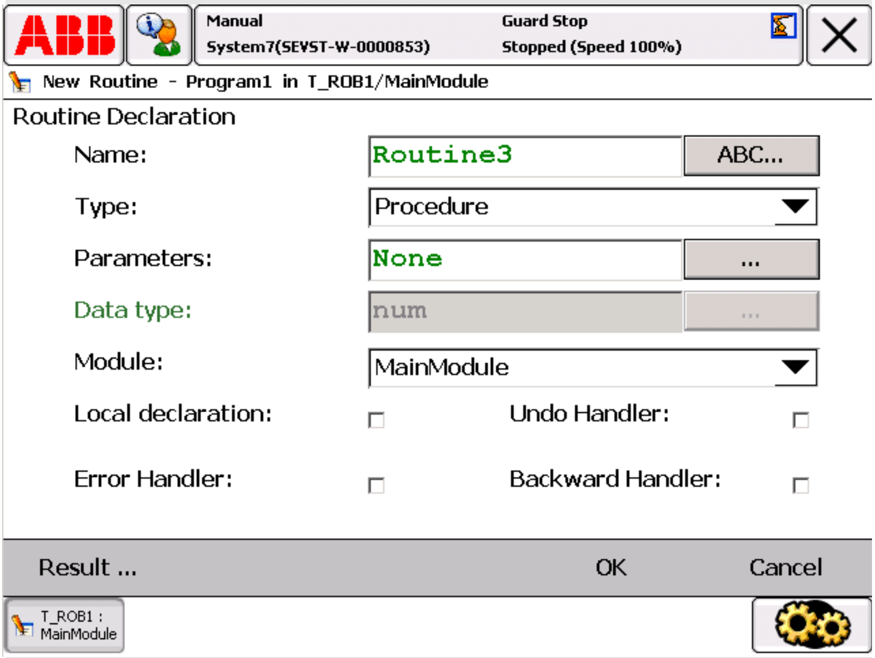
This section details how to handle program routines. i.e.:

- create a new routine
- create a copy of a routine
- change the declaration of a routine
- delete a routine

Creating a new routine

This section details how to create a new routine, set the declaration, and add it to a module.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Routines**.
3.	Tap **File**, then **New Routine**. A new routine is created and displayed with default declaration values.



en0400000692

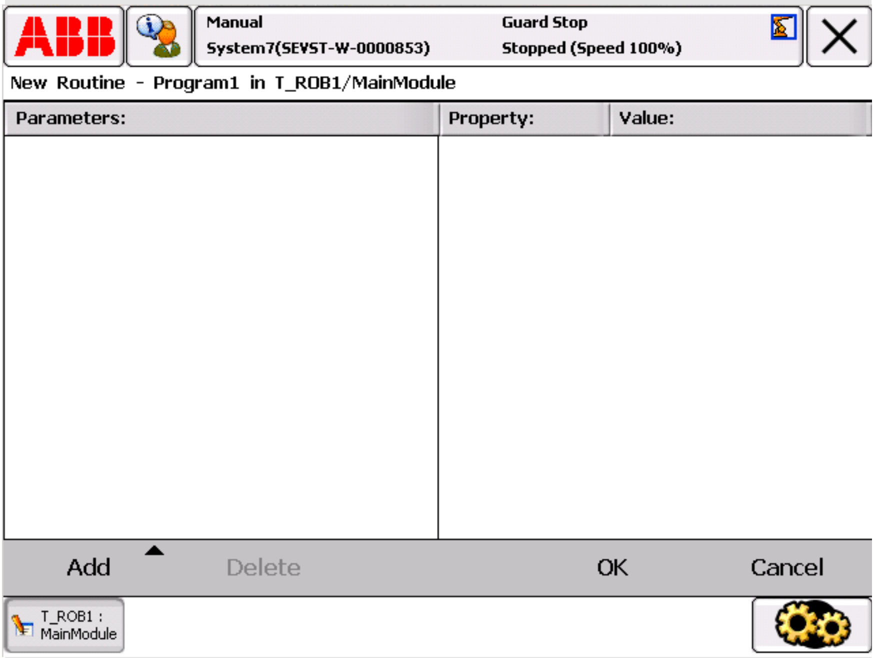
Step	Action
4.	Tap **ABC...** and use the soft keyboard to enter the new routines' name. Then tap **OK**.
5.	Select the type of routine: • Procedure: used for a normal routine without return value • Function: used for a normal routine with return value • Trap: used for an interrupt routine
6.	Do you need to use any parameters? If YES; tap **...** and proceed as detailed in section *Defining parameters in routine on page 173*. If NO; proceed to the next step.

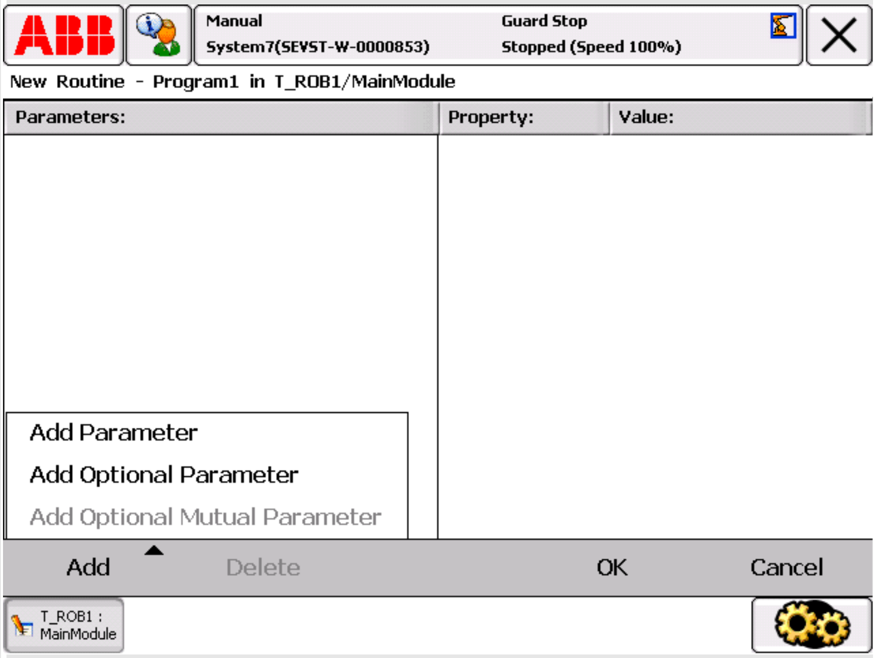
Continues on next page

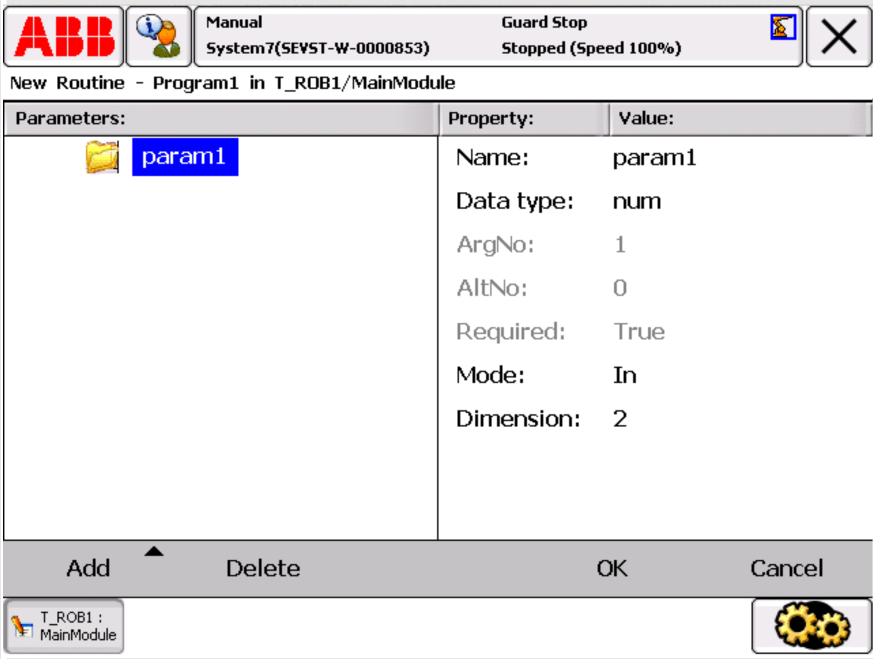
Step	Action
7.	Select module to add the routine to.
8.	Tap the checkbox to select **Local declaration** if the routine should be local. A local routine can only be used in the selected module.
9.	Tap **OK**.

Defining parameters in routine

This section describes how to define parameters in a routine.

Step	Action
1.	In the routine declaration, tap **...** to define parameters. A list of defined parameters is displayed.  en0400000693

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2.</td><td>If no parameters are shown, tap Add to add a new parameter.
Add optional parameter adds a parameter that is optional
Add optional mutual parameter adds a parameter that is mutually optional with another parameter
Read more about routine parameters in the RAPID reference manuals.


en0400000695</td></tr>
<tr><td>3.</td><td>Use the soft keyboard to enter the name of the new parameter and then tap OK.
The new parameter is displayed in the list.


en0400000696</td></tr>
<tr><td>4.</td><td>Tap to select a parameter. To edit values, tap the value.</td></tr>
</table>

Continues on next page

Step	Action
5.	Tap **OK** to return to the routine declaration.

Creating a copy of a routine

This section describes how to create a copy of a routine.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Routines**.
3.	Highlight the routine by tapping it.
4.	Tap **File**, then **Create copy**. The new routine is displayed. The name of the new routine is set to the same as the original with the suffix *Copy*.
5.	Make any changes in the declarations for the new routine copy. Then tap **OK**. How to make all declarations is detailed in section *Creating a new routine on page 172*.

Changing the declaration of a routine

This section describes how to change the declaration of a routine.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Routines**.
3.	Highlight the routine by tapping it.
4.	Tap **File**, then **Change Declaration**
5.	Change any declaration values for the routine. Then tap **OK**. Declaration settings are described in section *Creating a new routine on page 172*.

Deleting a routine

This section describes how to delete a routine from memory.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Routines**.
3.	Highlight the routine by tapping it.
4.	Tap **File**, then **Delete routine...** A dialog box is displayed.
5.	Tap: • **OK** to delete the routine without saving any changes made to it. • **Cancel** to revert without deleting the routine.

6.7.4. Handling of instructions

Instructions

A RAPID program consists of instructions. An instruction can, for example, move the robot, set an I/O signal, or write a message to the operator.

A large number of instructions are available, and these are listed in *RAPID reference manual - Instructions*. The basic procedure for adding instructions are, however, identical.

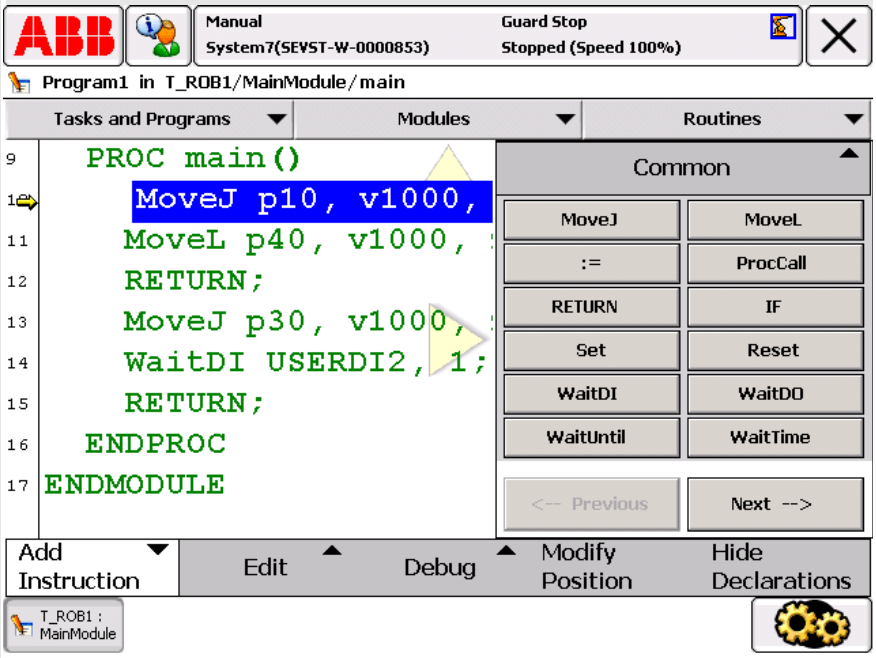
Undo and redo

When editing programs in the Program editor, you can undo and redo up to three steps. This function is available in the Edit menu.

Adding instructions

This section describes how to add instructions.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>On the ABB menu, tap Program Editor.</td></tr>
<tr><td>2.</td><td>Tap to highlight the instruction under which you want to add a new instruction.</td></tr>
<tr><td>3.</td><td>Tap Add instruction.
A category of instructions is displayed.



en0400000697
A large number of instructions, divided into several categories, are available. The default category is Common, where the most common instructions are listed.
You can create three personalized lists using the system parameters of the type Most Common Instruction in the topic Man-machine Communication. The system parameters are described in Technical reference manual - System parameters.</td></tr>
</table>

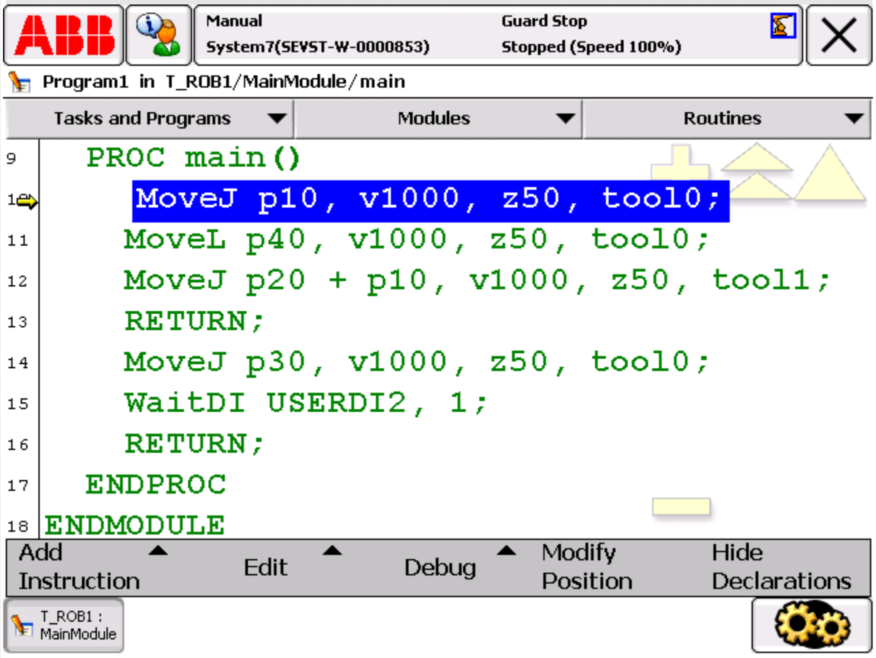
Continues on next page

Step	Action
4.	Tap **Common** to display a list of the available categories. You can also tap **Previous/Next** at the bottom of the list of instructions to move to the next/previous category.
5.	Tap the instruction you want to add. The instruction is added to the code.

Editing instruction arguments

This section describes how to edit instruction arguments.

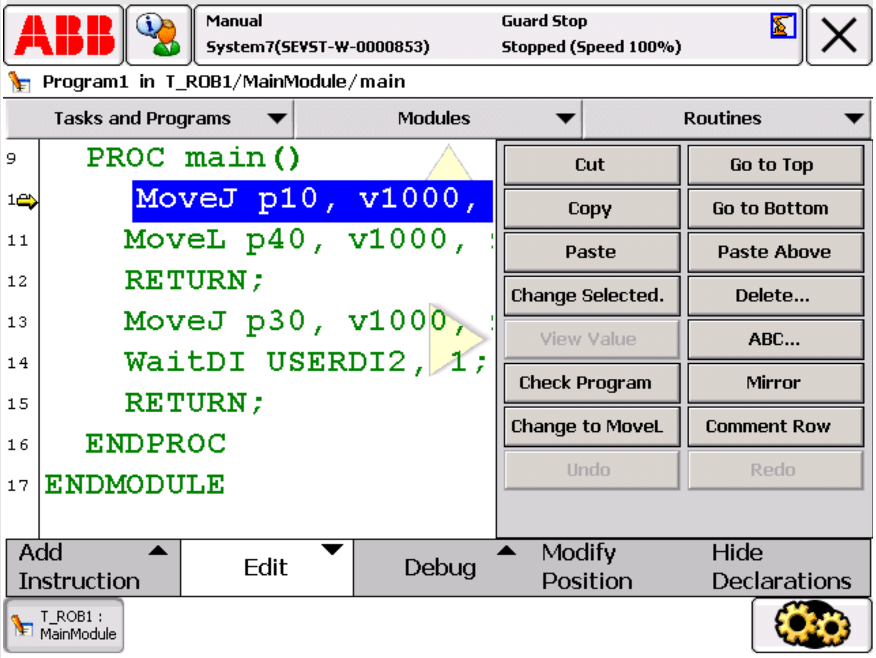
Step	Action
1.	Tap the instruction to edit.



en0400000699

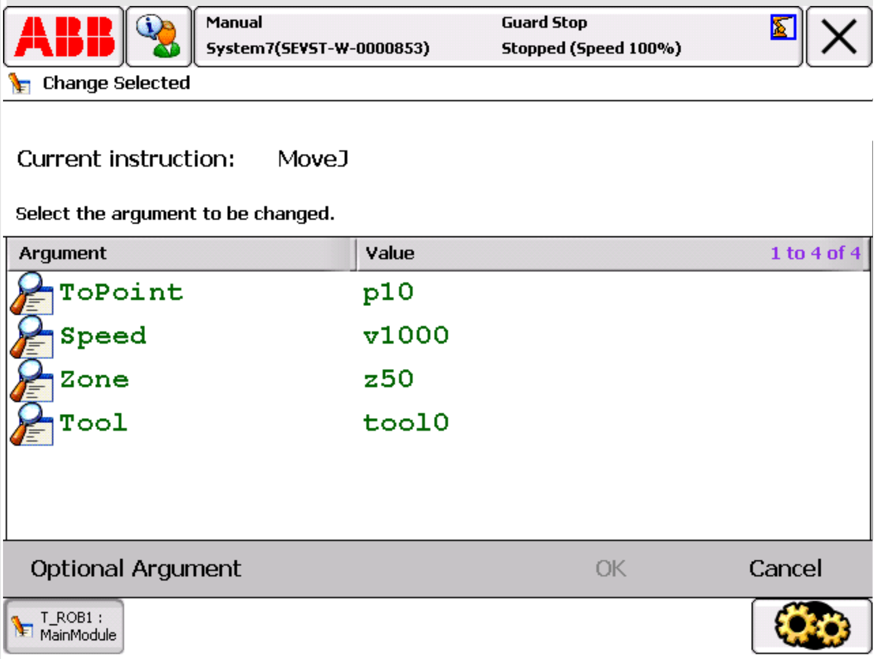
Continues on next page

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2.</td><td>Tap Edit.



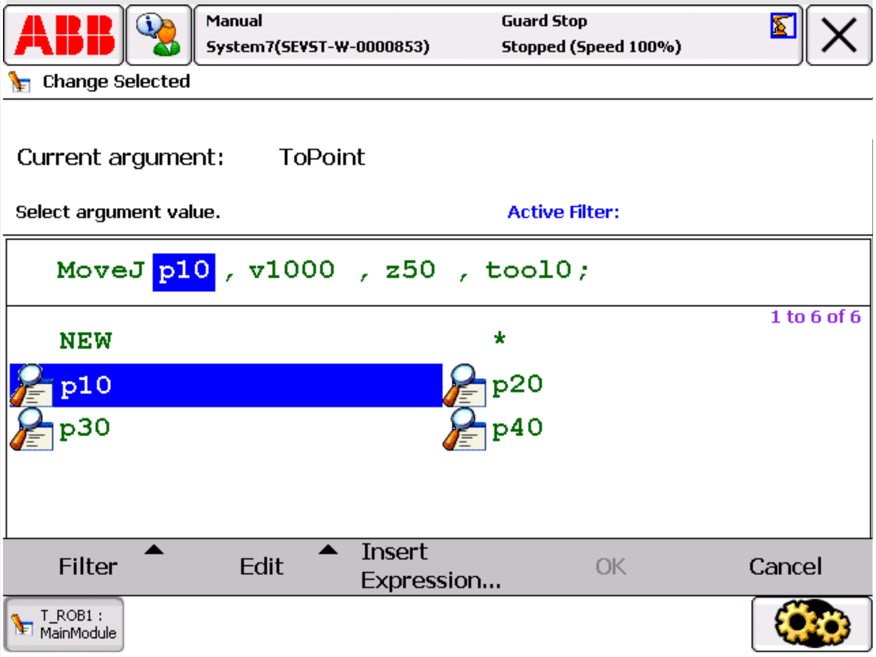
en0400000701</td></tr>
<tr><td>3.</td><td>Tap Change Selected.

Depending on the type of instruction, the arguments have different data types. Use the soft keyboard to change string values or proceed to the next steps for other data types or multiple argument instructions.



en0400000702</td></tr>
</table>

Continues on next page

Step	Action
4.	Tap the argument to be changed. A number of options are displayed.  en0400000703
5.	Tap an existing data instance to select and then tap **OK** to complete, or tap **Insert Expression...** See more about expressions in section *Editing instruction expressions and declarations on page 188*. To edit a particular data instance, see *Editing instruction expressions and declarations on page 188*.

TIP!

Tapping twice on an instruction will automatically launch the Change selected option. Tapping twice on an instruction argument will automatically launch the argument editor.

Copying and pasting instructions or arguments

This section describes how to paste instructions or arguments.

Step	Action
1.	Tap to select the argument or instruction you want to copy and then tap **Edit**.
2.	Tap **Copy**.
3.	Place the cursor on the instruction above where you want to paste the instruction or argument, or tap on the argument or instruction you want to change and tap **Paste**.

Continues on next page

Cutting an instruction

This section describes how to cut an instruction.

Step	Action
1.	Tap to select the instruction you want to cut and then tap **Edit**.
2.	Tap **Cut**.

Changing motion mode for a move instruction

This section describes how to change the motion mode for a move instruction.

Step	Action
1.	Tap to select the move instruction you want to change and then tap **Edit**.
2.	Tap **Change to Joint** or **Change to Linear**. The change is performed.

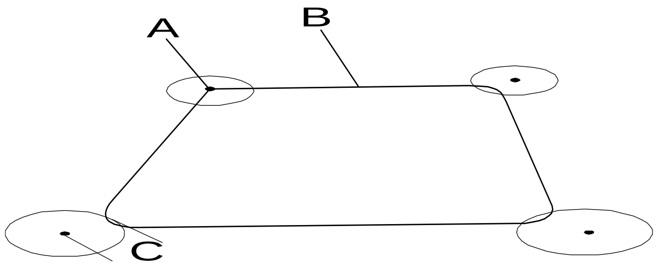
Commenting instruction rows

Instruction rows can be commented, i.e. skipped in the program execution. The comment/uncomment command is found under the Edit menu in the Program editor.

6.7.5. Example: Add movement instructions

Overview

In this example you will create a simple program that makes the robot move in a square. You need four movement instructions to complete this program.



en0400000801

A	First point
B	Robot movement Speed data v50 = speed 50mm/s
C	Zone z50 = (50mm)

Add movement instructions

This section details how to add movement instructions.

Step	Action	Info
1.	Jog the robot to the first point.	Tip: Use only left-right/up-down joystick movements to jog in a square.
2.	In the program editor, tap **Add Instruction**.	
3.	Tap **MoveL** to insert a `MoveL` instruction.	
4.	Repeat for the next three positions of the square.	
5.	Tap on z50 in the last instruction. Tap **Edit** and then **Change selected**.	
6.	Tap **Fine** to make the square end in a stop point. Tap **OK**.	

Result

Your program code should look like this:

```
Proc main()
  MoveL *, v50, z50, tool0;
  MoveL *, v50, z50, tool0;
  MoveL *, v50, z50, tool0;
  MoveL *, v50, fine, tool0;
End Proc;
```

6.8 Advanced programming

6.8.1. Mirroring a program, module, or routine

Mirroring

Mirroring creates a copy of a program, module, or routine with all positions mirrored in a specific mirror plane. In general, all data used in the original will be mirrored. Mirroring data only affects the initialization value, i.e. any current value will be ignored.

The mirror function can be applied to any program, module, or routine.

Mirroring a routine

This section describes how to mirror a routine.

Step	Action
1.	In the **ABB** menu, tap **Program Editor**.
2.	Tap **Edit** and tap **Mirror**.
3.	To define the mirror. • Tap the **Module** menu to select in which module the routine to mirror is used. • Tap the **Routine** menu to select which routine you want to mirror. • Tap **...** to open the soft keyboard and enter the name for the new routine.
4.	If you want to define the type of mirror then tap **Advanced options**, otherwise proceed to the next step. To define the type of mirror: • Deselect the **Base Mirror** checkbox. • Tap **...** to the right of **Work object** to select the work object frame to which all positions which are to be mirrored are related to. • Tap **...** to the right of **Mirror frame** to select the mirror plane to which all positions will be mirrored. • Tap the **Axis to mirror** menu to specify the mirroring orientation, x or y. • Tap **OK** to save the advanced options.
5.	Tap **OK**. A dialogue box is displayed.
6.	Tap **Yes** to apply the selected mirror to the routine, or tap **No** to cancel.

Mirroring a module or program

This section describes how to mirror a module or program.

Step	Action
1.	In the **ABB** menu, tap **Program Editor**.
2.	Tap **Edit** and tap **Mirror**.
3.	To define the mirror. • Tap the **Module** menu to select module to mirror. • Tap **...** to open the soft keyboard and enter the name for the new module or program.

Continues on next page

Step	Action
4.	If you want to define the type of mirror then tap **Advanced options**, otherwise proceed to the next step. To define the type of mirror: • Deselect the **Base Mirror** checkbox. • Tap **...** to the right of **Work object** to select the work object frame to which all positions which are to be mirrored are related to. • Tap **...** to the right of **Mirror frame** to select the mirror plane to which all positions will be mirrored. • Tap the **Axis to mirror** menu to specify the mirroring orientation, x or y. • Tap **OK** to save the advanced options.
5.	Tap **OK**. A dialogue box is displayed.
6.	Tap **Yes** to apply the selected mirror to the module, or tap **No** to cancel.

6.8.2. Modifying positions

Overview modifying positions

Positions are instances of the data type robtarget or jointtarget. See *RAPID reference manual - Functions and data types*.

The positions can be modified either using the function HotEdit, where you enter the new values using a soft keyboard, or using the Modify positions function in the Program editor where you step and jog the robot to the new position.

HotEdit is described in section *HotEdit on page 73*.

Note that jointtargets can only be modified using the Modify positions method in the Program editor, i.e. not with HotEdit.

CAUTION!

Changing programmed positions may significantly alter the robot's movement pattern.

Always make sure any changes are safe for both equipment and personnel.

Overview modifying positions in the Program editor

When modifying positions by jogging the robot to the new position you can either single-step through the program to the position(s) you want to modify, or jog directly to the new position and change the corresponding position argument of the instruction.

The recommendation is to step to through the program to the position, but if you know your robot program well and the new position is known, it is faster to use the jogging method. Note! Do not use this method to change orientation values.

To modify positions using the Program editor, the system must be in manual mode.

Modifying positions

This procedure describes how to modify positions, either by single-stepping to the positions or jogging.

Step	Action	Info
1.	On the **ABB** menu, tap **Program Editor**.	
2.	Stop the program, if running.	
3.	Do you want to single-step to the position or jog? If *single-stepping*, step through the program to the position you want to change. Make sure the correct argument is selected. If *jogging*, use the **Jogging** view to make sure that the same work object and tool that are used in the instruction are selected.	When single-stepping, if the instruction or procedure call has more than one position argument, continue to step to reach each argument.
4.	Jog to the new position.	
5.	When using the jogging method, tap to select the position argument you want to change.	
6.	Tap **Modify Position**. A confirmation dialog appears.	
7.	Tap **Modify** to use the new position, **Cancel** to keep the original.	

Continues on next page

Step	Action	Info
8.	Repeat step 3 through 7 for each position argument you want to change.	
9.	Tap **Close** to close the Program Editor.	

Limitations

The **Modify Position** button in the Program editor is disabled until you select a position argument.

The maximum movement or change in orientation, may be restricted by the system parameters (topic *Controller*, type *ModPos Settings*) in the system design. Please read your cell or plant documentation for details.

If the system parameters are setup to use absolute limits for position changes, then the original positions can only be restored or changed using the baseline menu in HotEdit.

If a named position is changed, all other instructions using that position will be affected.

Related information

HotEdit on page 73.

RAPID reference manual - Functions and data types.

Technical reference manual - System parameters.

6.8.3. Moving the robot to a programmed position

Positions

A robot program usually contain programmed positions. The robot can move automatically to a programmed position using a function in the Jogging menu.

The robot will move at 250 mm/s.

DANGER!

When moving the robot automatically, the robot arm may move without warning. Make sure no personnel are in safeguarded space and that no objects are in the way between the current position and the programmed position.

Moving the robot to a programmed position

This procedure describes how to move a robot automatically to a programmed position.

Step	Action	Info
1.	On the **ABB** menu, tap **Jogging**.	
2.	Make sure the correct mechanical unit is selected and then tap **Go To...**.	
3.	Tap to select a programmed position.	If you have many programmed positions you can use a filter to narrow down the visible positions. See section *Filtering data on page 101*.
4.	Press and hold the enabling device and then tap and hold the **Go To** button. The robot now moves directly from the current position to the programmed position. Make sure no objects are in the way.	

6.8.4. Aligning tools

Overview

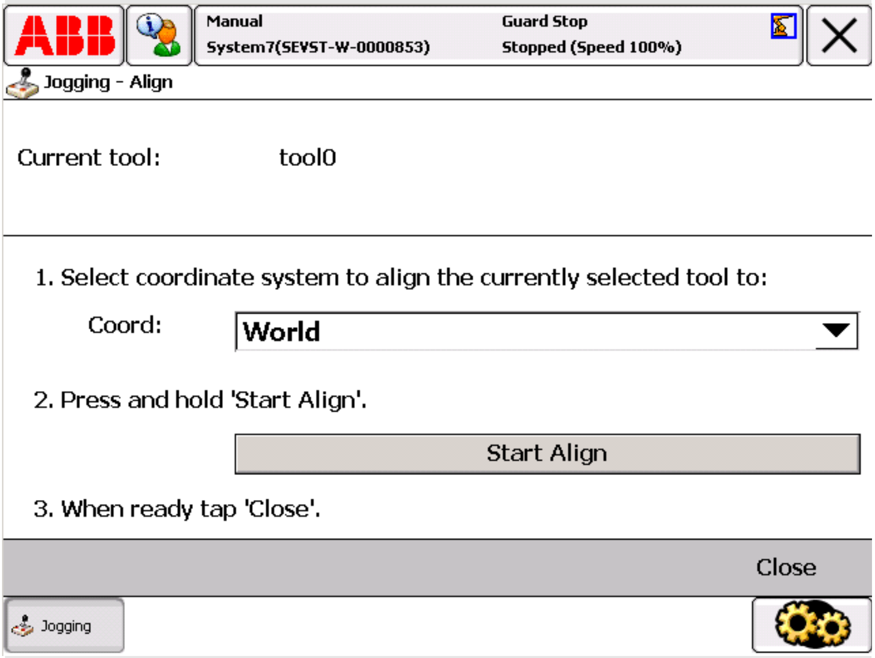
A tool can be aligned with another coordinate system.

When aligning a tool, the tool's z axis is aligned to the selected coordinate system's nearest axis. Therefore it is recommended to first jog the tool so it is close to the desired coordinates.

Note that the tool's data is not changed!

Aligning mechanical units

This procedure describes how to align tools.

Step	Action
1.	On the **ABB** menu, tap **Jogging**.
2.	Make sure that the right tool is active and then tap **Align...**.  en0500001548
3.	Select a coordinate system to align the the selected tool to.
4.	Press and hold the enabling device and then tap and hold **Start Align** to start aligning the tool.
5.	Tap **Close** when completed.

6.8.5. Editing instruction expressions and declarations

Expressions

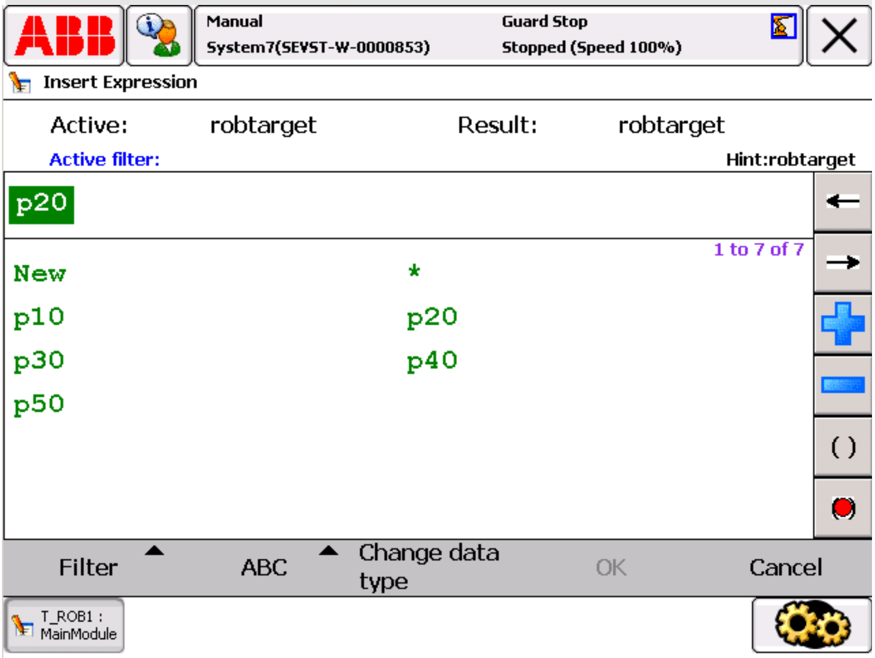
An expression specifies the evaluation of a value. It can be used, for example:

- as a condition in an IF instruction
- as an argument in an instruction
- as an argument in a function call

Read more in *RAPID reference manual - RAPID overview* and *RAPID reference manual - Instructions*.

Inserting expressions

This procedure describes how to insert and edit expressions in instructions.

Step	Action
1.	In the **Program Editor**, tap to select the instruciton you want to edit and then tap **Edit**.
2.	Tap **Change selected** and tap to select the argument to change.
3.	Tap **Insert expression**.  en0400000704
4.	Edit the length of the expression by tapping the keys to the right: • Arrows: step backward and forward in the expression. • + to add expression. Tap the new expression to define it. • - to delete expression. • () to set a parenthesis around the highlighted expression. • (o) to delete a parenthesis.

Continues on next page

Step	Action
5.	Tap: • **New** to create a new data declaration, i.e. adding a data declaration not previously used. This is detailed in section *Creating new data declarations on page 189*. • **View** to change views or change data type. This is detailed in section *Changing data type on page 190*. • **ABC** displays the soft keyboard.
6.	Tap **OK** to save the expression.

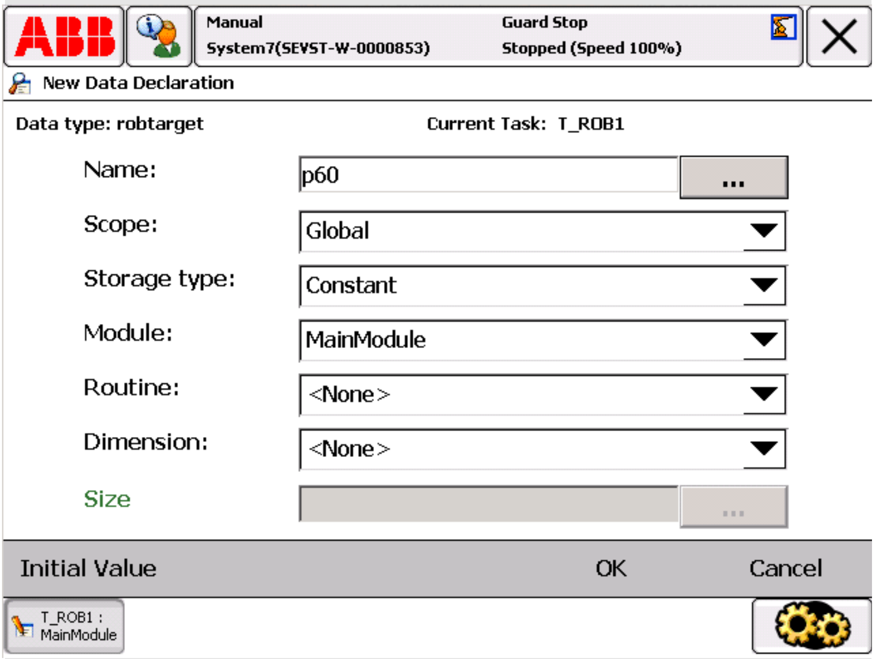
Declarations and data types

When editing an expression new data can be declared with the button **New**. More information about data declarations and how to edit them can be found in section *Editing data instances on page 139*.

Creating new data declarations

This procedure describes how to create a new data declaration in an instruction expression.

Step	Action
1.	In the Insert Expression view, tap **New**.



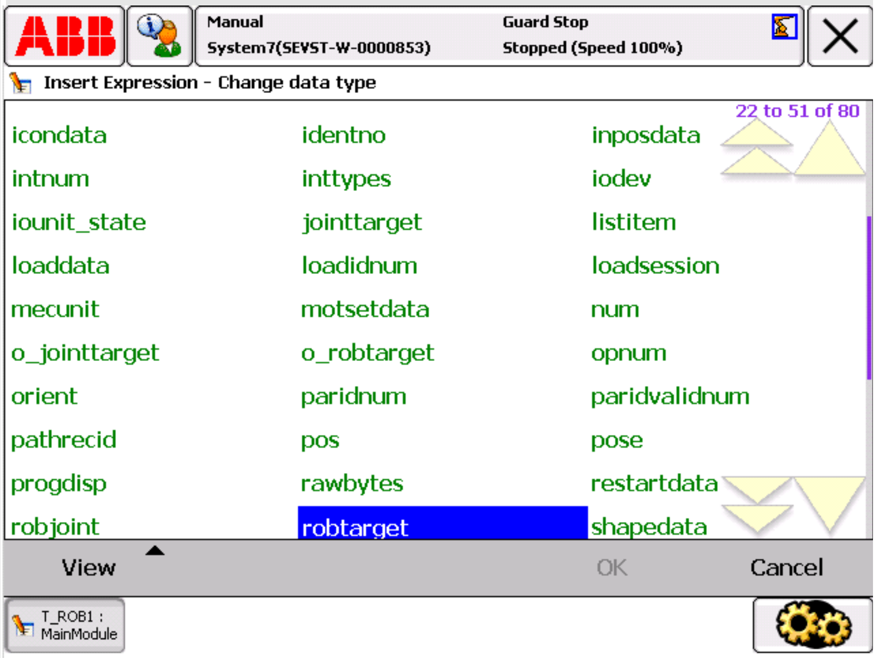
en0400000705

Continues on next page

Step	Action
2.	Tap and enter desired values: • **Initial value** to set the initial value. • **...** to display the soft keyboard and change the data type's name. • **Scope** • **Storage type** • **Module** • **Routine** • **Dimension** to set the size of an array if the data type should be an array. • If a value has been chosen for Dimension, tap **...** to set array size, see *What is a data array? on page 312*
3.	After making all selections, tap **OK**. A dialog box is displayed, prompting you to allow resetting of the program pointer and applying all changes: • Tap **Yes** to proceed. • Tap **No** to return to the data type view without resetting of the program pointer or applying changes.

Changing data type

This section describes how to change data type.

Step	Action
1.	In the Insert Expression view, tap **View** and then **Change data type**, the following screen is displayed:  en0400000706
2.	Tap to select the required data type and tap **OK**.

6.8.6. Hiding declarations in program code

Declarations

Program declarations can be hidden to make the program code easier to read.

Hiding declarations

This section describes how to hide or show declarations.

Step	Action
1.	In the **ABB** menu, tap **Program Editor** to view a program.
2.	Tap **Hide Declarations** to hide declarations. Tap **Show Declarations** to show declarations.

6.8.7. Deleting programs from memory

Overview

Deleting a program in a task does not erase the program from the controller hard disk but only from the program memory.

When you switch programs, the previously used program is deleted from the program memory, but not removed from the hard disk if it was saved there.

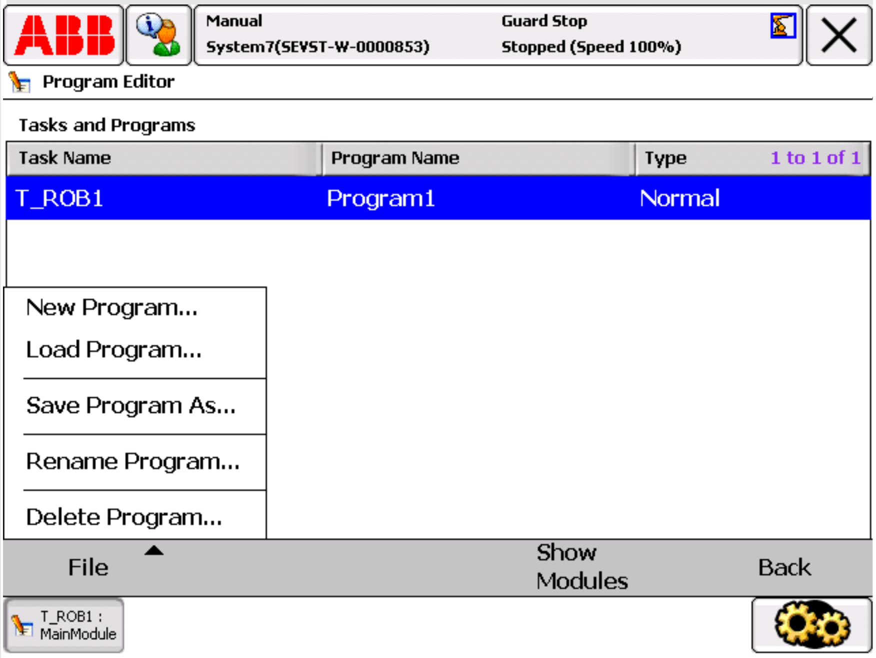
How to save your work is detailed in section *Handling of programs on page 165*.

The different memories are described in section *What is "the memory"? on page 248*.

Deleting programs from memory

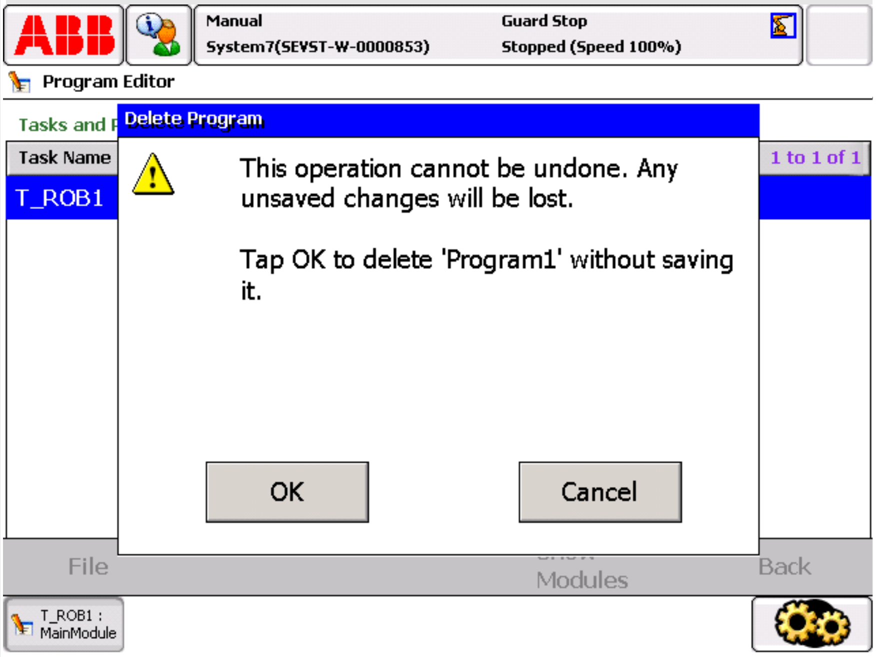
This section details how to delete programs from the program memory.

Step	Action
1.	On the ABB menu tap **Program Editor**.
2.	Tap **Tasks and Programs**.
3.	Tap **File**.



en0400000678

Continues on next page

Step	Action
4.	Tap **Delete Program...**. **WARNING!** Recent program changes will not be saved.  en0400000853
5.	Tap **OK**. If you don't want to lose information about program changes then use **Save Program** before deleting the program. How to save your work is described in section *Handling of programs on page 165*.

6.8.8. Deleting programs from hard disk

Overview

Programs are deleted via FlexPendant Explorer or an FTP client. When deleting programs from the controller hard disk, the currently loaded program in the program memory is not affected.

The different memories are described in section *What is “the memory”? on page 248*.

Deleting programs with FlexPendant Explorer

Programs can be deleted using FlexPendant Explorer on the ABB menu. See section *FlexPendant Explorer on page 75*.

6.8.9. Activating mechanical units

Overview

A mechanical unit can be active or deactive. Only active units are run when executing a program. Deactivated units will not run. This may be useful when programming or testing a program.

A robot **cannot** be deactivated.

The Activate function does not affect jogging. To select mechanical unit for jogging, use the Mechanical unit property in the Jogging menu.

Activating mechanical units

This procedure describes how to activate a mechanical unit.

Step	Action	Info
1.	On the **ABB** menu, tap **Jogging**.	
2.	Make sure that the right mechanical unit is selected, then tap **Activate...**. To deactivate an active mechanical unit, tap **Deactivate**.	A robot cannot be deactivated.

Related information

Selecting mechanical unit on page 108.

Mechanical units can be active or deactive at startup depending on the system setup, see *Technical reference manual - System parameters*, topic *Motion*.

6.9 Testing

6.9.1. How to use the hold-to-run function

When to use the hold-to-run function

The hold-to-run function is used to run or step programs in manual full speed mode, in combination with the enabling device.

In systems with hold-to-run buttons on the back of the FlexPendant, one of these buttons are used. In systems without hold-to-run buttons, use the Start, Forward, or Backward buttons instead (press and hold). The hold-to-run function is the same.

Operational mode	Function
Manual reduced speed mode	Normally, hold-to-run has no effect in the manual reduced speed mode. However, it is possible to activate for manual reduced speed mode by changing a system parameter.
Manual full speed mode	Pressing hold-to-run AND pressing the enabling device enables running a program. It may be run continuously or step-by-step. Releasing hold-to-run in this mode immediately stops manipulator movement as well as program execution. When pressing it again, execution is resumed from that position.
Automatic mode	Hold-to-run is not used in automatic mode.

How to use the hold-to-run function

This instruction details how use the hold-to-run function in manual full speed mode.

Step	Action
1.	Press the enabling device on the FlexPendant.
2.	In a system *with* hold-to-run buttons, choose execution mode by pressing either: • **Start** (continuous program execution) • **Forward** (step-by step program execution forwards) • **Backward** (step-by step program execution backwards) Then wait for the hold-to-run alert box and press the hold-to-run button. You can also select step mode, see section *Stepping instruction by instruction on page 205*.
3.	In a system *without* hold-to-run buttons, choose execution mode by pressing and holding either: • **Start** (continuous program execution) • **Forward** (step-by step program execution forwards) • **Backward** (step-by step program execution backwards)
4.	If **Start** was pressed, then the program execution continues as long as the hold-to-run or Start button is pressed. If **Forward** or **Backward** was pressed, the program is executed step-by-step by alternately releasing and pressing the hold-to-run or the Forward/Backward button.

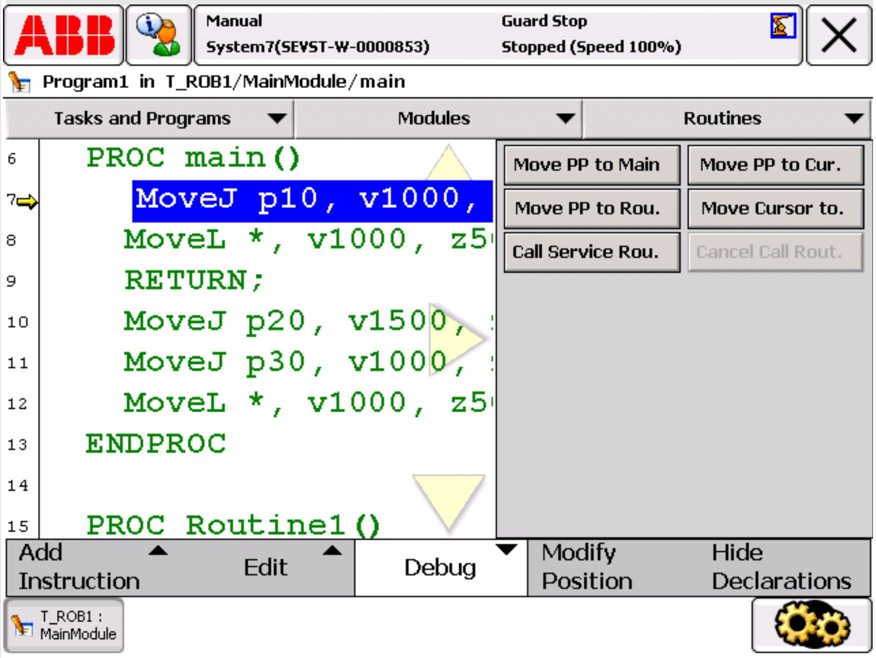
Continues on next page

Step	Action
5.	If the hold-to-run button is released, program execution stops. If the hold-to-run button is pressed again after being released, program execution is resumed from the position in which it was released.
6.	It is possible to change execution mode when the hold-to-run button is released and then continue the program execution with the new execution mode, by just activating the hold-to-run button again.
7.	If the enabling device is released, intentionally or by accident, the complete procedure must be repeated to enable running.

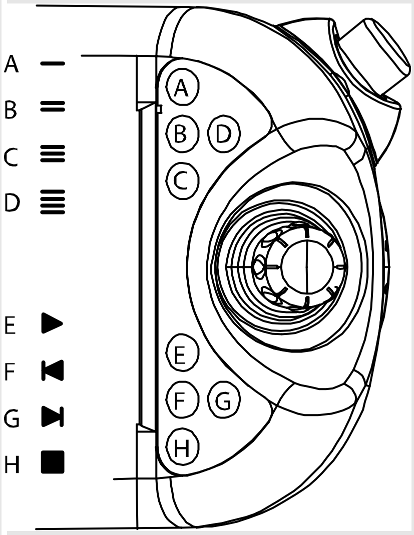
6.9.2. Running the program from a specific instruction

Running the program from a specific instruction

This section details how to run the program from a specific instruction.

Step	Action
1.	On the **ABB** menu Tap **Program Editor.**
2.	Tap **Debug**.  en0400000868 Tap on the program step where you want to start, then tap **Debug** and select **Move PP to Cur**.

Continues on next page

Step	Action
3.	 xx **Make sure that no personnel are in the robot working area.** **WARNING!** Before running the robot, observe the safety information in section *DANGER - Moving manipulators are potentially lethal! on page 18*.
4.	Press the **Start** button on the FlexPendant hardware button set (see E in illustration below).  en0300000587

6.9.3. Running a specific routine

Overview

To run a specific routine you must have the module with the routine loaded. How to run service routines is described in section *Running a service routine on page 209*.

Running a specific routine

This procedure describes how to run a specific routine in manual mode.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Debug** and then **Move PP to routine** to place the program pointer at the start of the routine.
3.	Press the Start button on the FlexPendant.

6.9.4. Quickset menu, Run Mode

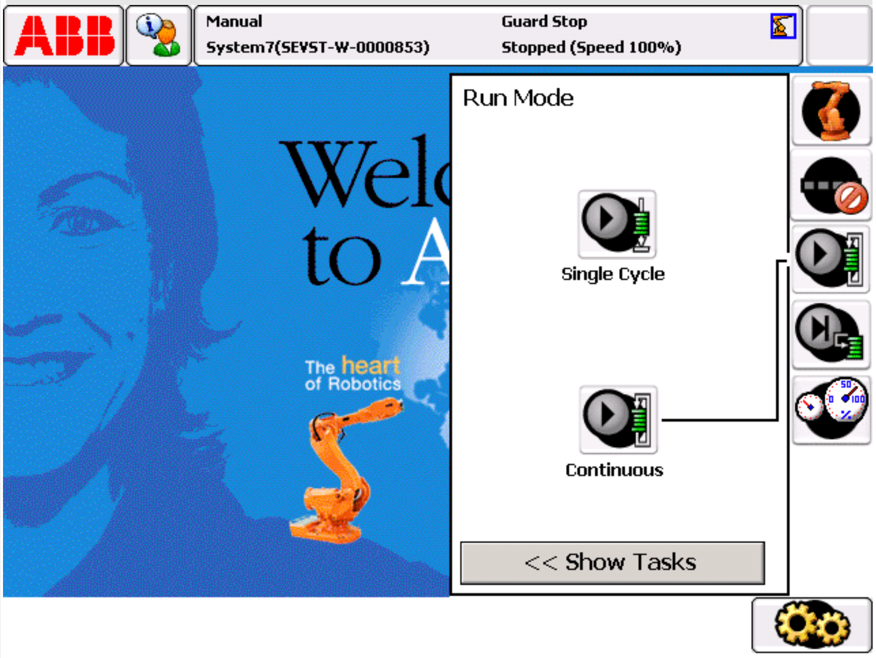
Run mode

By setting run mode you define if the program execution should run once and then stop, or run continuously.

Under the Run mode menu, you can also select which tasks should be active or deactivated if you have the Multitasking option installed. Select task is also available under the Quickset Step mode menu.

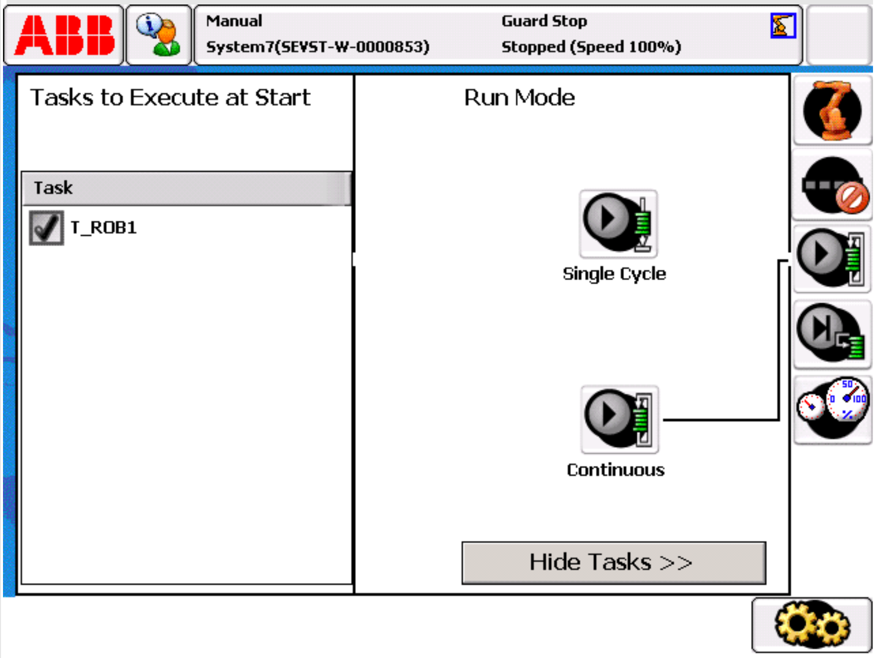
Select run mode

This section describes how to select run mode using the Quickset menu.

Step	Action
1.	If you want to view/change any Run mode functionality, tap the Run Mode button.  en0300000472 The following buttons are displayed: • Single cycle running • Continuous running The Show tasks button opens the list with available tasks.

Continues on next page

Step	Action
2.	If you want to view/change which tasks that are active, tap the Show Tasks button. To select tasks you must be in manual mode.



en0400000992

The following is displayed:

- A list of all available tasks. Tap to activate or deactivate tasks.
- Hide tasks will hide the task list.

6.9.5. Quickset menu, Step Mode

Step mode

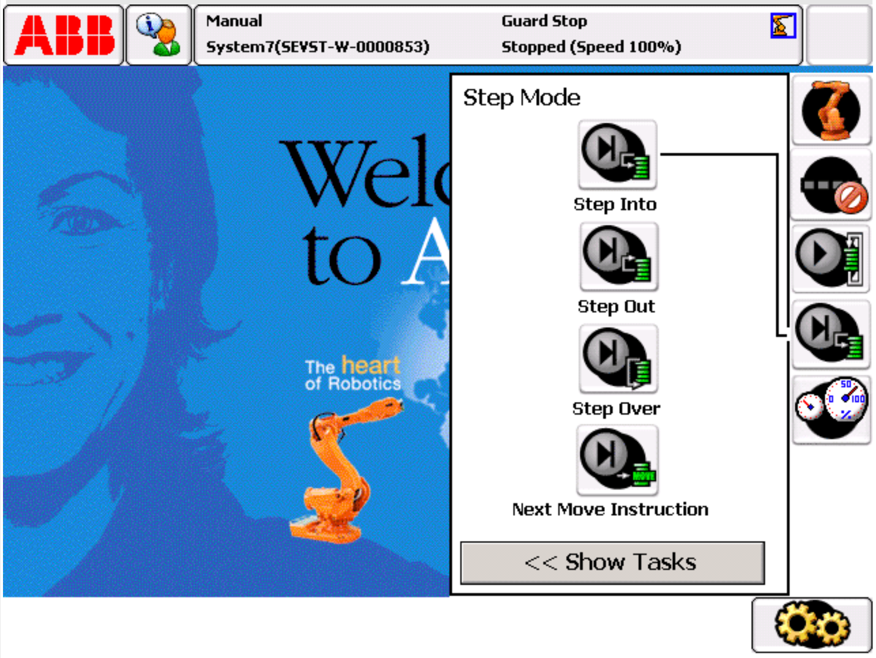
By setting step mode you define how the step-by-step program execution should function.

Under the Step mode menu, you can also select which tasks should be active or deactivated if you have the Multitasking option installed. Select task is also available under the Quickset Run mode menu.

Select step mode

This section describes how to select step mode using the Quickset menu.

Step	Action
1.	If you want to view/change any Step mode functionality, tap the Step Mode button.



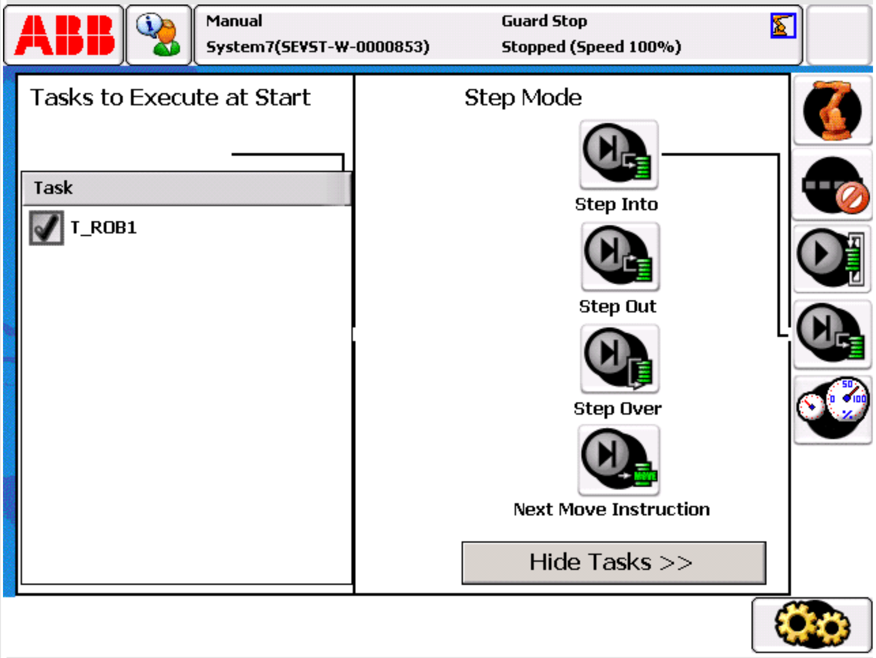
en0300000543

After tapping **Step Mode**, the following buttons are displayed:

- Step into. Steps into called routines and executes them step-by-step.
- Step out. Executes the remains of the current routine and then stops at the next instruction in the routine from which the current routine was called. Not possible to use in the Main routine.
- Step over. Called routines are executed in one single step.
- Next move instruction. Steps to the next move instruction. Stops before and after movement instructions to allow e.g. modifying positions.

Show tasks opens the list of available tasks.

Continues on next page

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2.</td><td>If you want to view/change which tasks that are active, tap the Show Tasks button. To select tasks you must be in manual mode.


en0400000993
The following is displayed:
• A: A list of all available tasks. Tap to activate or deactivate tasks.
• B: Hide tasks will hide the task list.</td></tr>
</table>

Related information

Modifying positions on page 184.

6.9.6. Stepping instruction by instruction

Overview

In all operating modes the program may be executed step by step forwards or backwards.

Stepping backwards is limited, see *RAPID overview* for more details.

Select step mode

This section details how to select step mode. Stepping can be done in three ways; step in, step over, and motion step.

Step	Action	Info
1.	Select step mode using the Quickset menu.	Described in *Quickset menu, Step Mode on page 203*.

Stepping

This section details how to step forwards and backwards.

If you want to step...	then press...
forward	Forward button on FlexPendant
backward	Backward button on FlexPendant

Limitations of backward execution

There are some restrictions for the backward execution:

- When stepping backwards through a `MoveC` instruction, the execution does not stop in the circular point.
- It is not possible to step backwards out of a `IF`, `FOR`, `WHILE` and `TEST` statement.
- It is not possible to step backwards out of a routine when reaching the beginning of the routine.
- There are instructions affecting the motion that cannot be executed backwards (e.g. `ActUnit`, `ConfL` and `PDispOn`). If attempting to execute these backwards, an alert box will inform you that this is not possible.

Backward execution behavior

When stepping forward though the program code, a program pointer indicates the next instruction to execute and a motion pointer indicates the move instruction that the robot is performing.

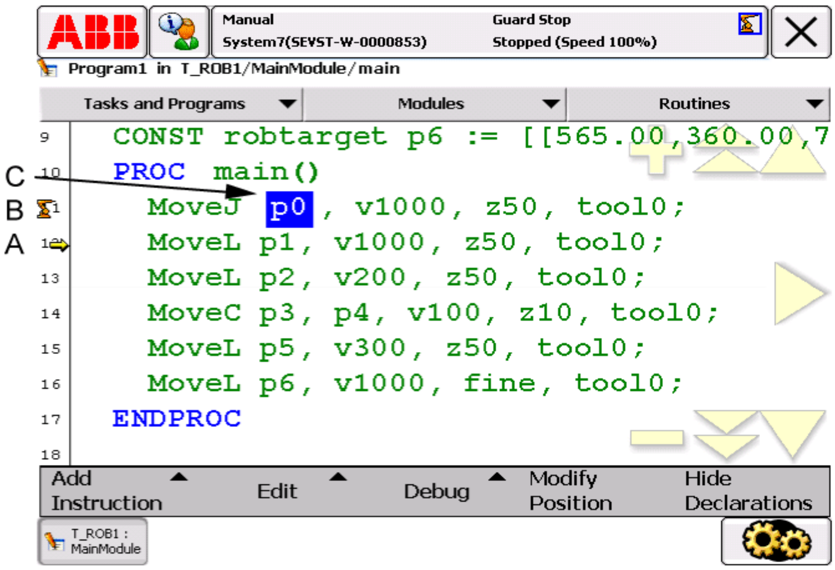
When stepping backward though the program code, the program pointer indicates the instruction above the motion pointer. When the program pointer indicates one move instruction and the motion pointer indicates another, the next backward movement will move to the target indicated by the program pointer, using the type of movement and speed indicated by the motion pointer.

Continues on next page

Example of backward execution

This example illustrates the behavior when stepping backwards through move instructions. The program pointer and motion pointer helps you keep track of where the RAPID execution is and where the robot is.

`MoveL`, `MoveJ`, and `MoveC` are move instructions in RAPID, see *RAPID reference manual - Instructions*.



en0400001204

A	Program pointer
B	Motion pointer
C	Highlighting of the robtarget that the robot is moving towards, or already has reached.

When...	then...
stepping forward until the robot is in p5	the motion pointer will indicate p5 and the program pointer will indicate the next move instruction (`MoveL p6`).
pressing the Backward button once	the robot will not move but the program pointer will move to the previous instruction (`MoveC p3, p4`). This indicates that this is the instruction that will be executed the next time Backward is pressed.
pressing the Backward button again	the robot will move to p4 linearly with the speed v300. The target for this movement (p4) is taken from the `MoveC` instruction. The type of movement (linear) and the speed are taken from the instruction below (`MoveL p5`). The motion pointer will indicate p4 and the program pointer will move up to `MoveL p2`.
pressing the Backward button again	the robot will move circularly, via p3, to p2 with the speed v100. The target p2 is taken from the instruction `MoveL p2`. The type of movement (circular), the circular point (p3) and the speed are taken from the `MoveC` instruction. The motion pointer will indicate p2 and the program pointer will move up to `MoveL p1`.

Continues on next page

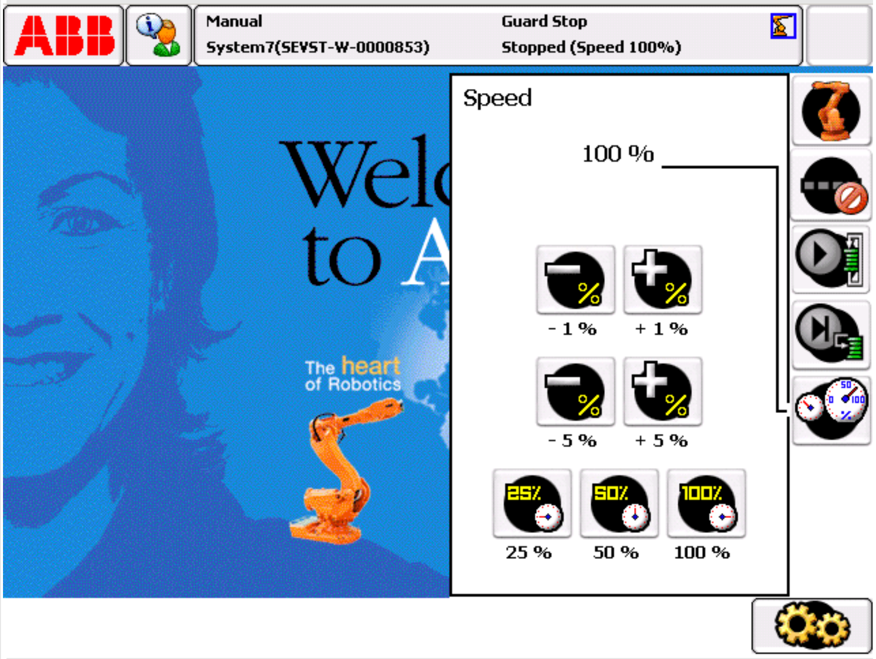
When...	then...
pressing the Backward button again	the robot will move linearly to p1 with the speed v200. The motion pointer will indicate p1 and the program pointer will move up to `MoveJ p0`.
pressing the Forward button once	the robot will not move but the program pointer will move to the next instruction (`MoveL p2`).
pressing the Forward button again	the robot will move to p2 with the speed v200.

6.9.7. Quickset menu, Speed

Speed button

The speed settings apply to the current operating mode. However, if you decrease the speed in automatic mode, the setting also applies to manual mode if you change mode.

Step	Action
1.	If you want to view/change any Speed functionality, tap the Speed button.



en0300000470

The following buttons are displayed (from top left):

- Actual running speed (in relation to max)
- Decrease running speed in steps of 1%
- Increase running speed in steps of 1%
- Decrease running speed in steps of 5%
- Increase running speed in steps of 5%
- Run at quarter speed (25%)
- Run at half speed (50%)
- Run at full speed (100%)

6.10 Service routines

6.10.1. Running a service routine

Service routines

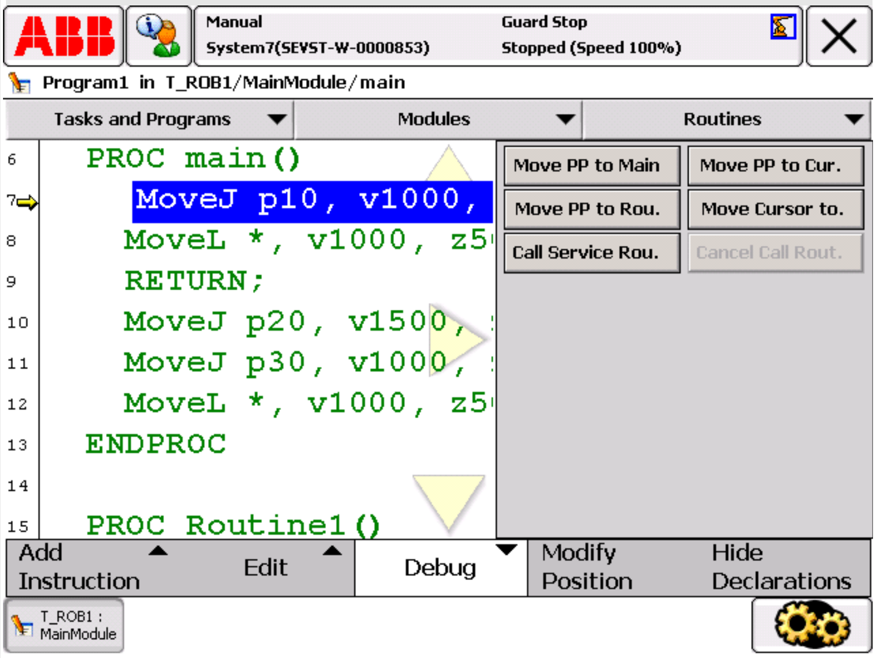
Service routines are routines for performing a number of common services. The service routines available depend on your system setup and available options. Please refer to your plant or cell documentation for more information.

Note that once a service routine has started running, aborting might not resume the system to previous state since the routine may have moved the robot arm.

Running a service routine

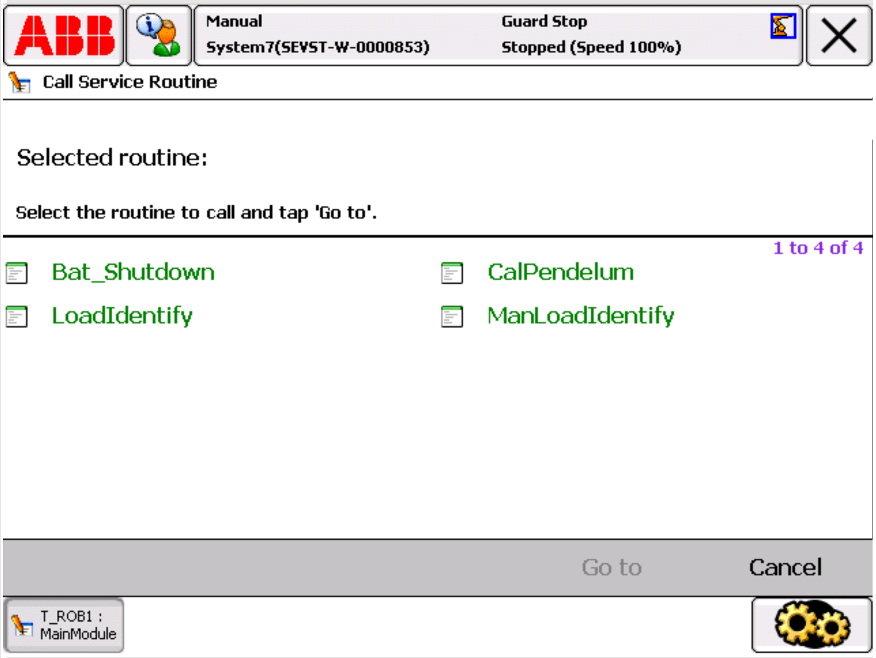
This section describes how to run a service routine in manual mode.

Step	Action
1.	On the **ABB** menu tap **Program Editor**.
2.	Tap **Debug**.



en0400000868

Continues on next page

Step	Action
3.	Tap **Call Service Routine** to list the service routines. Tap the routine you want to run and tap **Go to**.  en0400000885
4.	If the routine was loaded properly, press Start on the FlexPendant. The routine is now started and run. Follow any instructions on the screen.

Related information

Battery shutdown service routine on page 211.

LoadIdentify, load identifiction service routine on page 214.

Service Information System, ServiceInfo service routine on page 213.

Calibration Pendulum, CalPendulum service routine on page 212.

6.10.2. Battery shutdown service routine

Bat_shutdown

It is possible to shutdown the battery backup of the Serial Measurement Board to save battery power during transportation or storage. The function is reset when the system is powered on again. The revolution counters will be lost and needs an update but the calibration values will remain.

Related information

How to start a service routine is described in *Running a service routine on page 209*.

The Serial Measurement Board is described in *Serial Measurement Board memory on page 280*.

How to update the revolution counters is described in *Updating revolution counters on page 274*.

6.10.3. Calibration Pendulum, CalPendulum service routine

CalPendulum

CalPendulum is a service routine used with Calibration Pendulum, the standard method for calibration of ABB robots. This is the most accurate method for the standard type of calibration, and it is also the recommended method in order to achieve proper performance.

The calibration equipment for Calibration Pendulum is delivered as a complete toolkit, including the manual *Calibration Pendulum instruction*.

Related information

Running a service routine on page 209.

Calibration Pendulum is described in full in the manual *Calibration Pendulum instruction*. Specific information for each robot is described in the robot's product manual.

6.10.4. Service Information System, ServiceInfo service routine

ServiceInfo

ServiceInfo is a service routine based on Service Information System, SIS, a software function which simplifies maintenance of the robot system. It supervises the operating time and mode of the robot, and alerts the operator when a maintenance activity is scheduled.

Maintenance is scheduled by setting the system parameters of the type *SIS parameters*. How to work with system parameters is described in section *Configuring system parameters on page 270*. All system parameters are described in *Technical reference manual - System parameters*. See also the product manual for the robot.

Supervised functions

The following counters are available:

- Calender time counter
- Operation time counter
- Gearbox operation time counters

Counters are reset when maintenance has been performed.

The counter status is displayed after running the ServiceInfo routine for maintenance. Status “OK” indicates that no service interval limit has been exceeded by that counter.

Related information

Running a service routine on page 209.

Configuring system parameters on page 270.

The system parameters for SIS are described in *Technical reference manual - System parameters*, chapter *Motion*.

6.10.5. LoadIdentify, load identifiction service routine

Overview

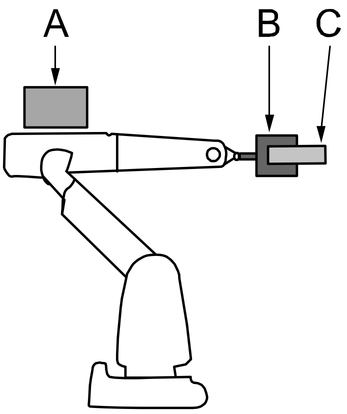
The service routine LoadIdentify is used to automatically calculate the data of loads mounted on the robot. You can also enter the data manually, but this requires information that may be difficult to calculate.

If you want to run load identification for the payload, make sure that the tool is correctly defined first, e.g. by running LoadIdentify for the tool.

To run LoadIdentify, there are a number of things to consider. These are described on the following pages. You can also find information on error handling and limitations in this chapter.

LoadIdentify

LoadIdentify can identify the tool load and the payload. The data that can be identified are mass, center of gravity, and moments of inertia. Together with the identified data a measurement accuracy, indicating how well the identification went, is also given.



en0500001535

A	Upper arm load
B	Tool load
C	Payload

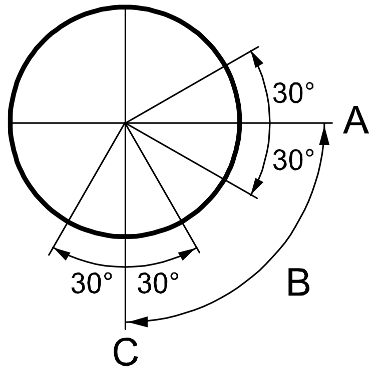
The movements of axis 3 will only be performed if the mass is to be identified. This means that to identify the mass, the upper arm load must be known and correctly defined first.

Calibration angles

To perform the identification the robot moves the load after a specific pattern and calculates the data. The axes that move are 3, 5 and 6. At the identification position, the motion for axis 3 is approximately 3 degrees up and 3 degrees down and for axis 5 it is approximately 30 degrees up and 30 degrees down. For axis 6 the motion is performed around two configuration points.

The optimum value for the configuration angle is 90 degrees.

Continues on next page

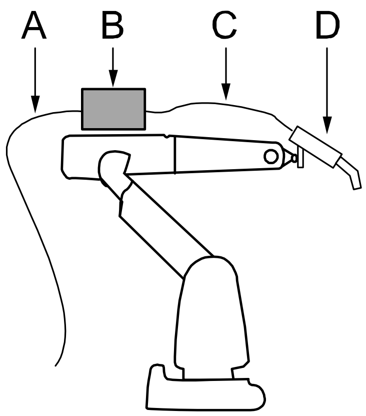


en0500001537

A	Configuration 2
B	Configuration angle
C	Configuration 1 (start position)

LoadIdentify with arm loads mounted

The best way to perform load identification is with a robot with no arm loads mounted. If this is not possible, good accuracy can still be achieved. Consider, for example, the robot in the figure below, which has arc welding equipment mounted on it.



en0500001536

A	Cable 1
B	Load 1
C	Cable 2
D	Load 2

If we want to use load identification to find the data of load 2, the most important thing to remember is to make sure that the upper arm load is correctly defined, in particular its mass and its center of gravity along the robot arm. The arm load includes everything that is mounted on the robot, except tool load and payload. In the figure above, cable 1, cable 2, and load 1 are included in the arm load.

Continues on next page

When performing the load identification, cable 2 should be disconnected since it will otherwise put an extra force on load 2. When identifying load 2 with such a force present, the result may differ considerably from the correct load. Ideally, cable 2 should be disconnected from load 2 and fastened on the upper arm. If this is not possible, the cable can also be disconnected at load 1 and fastened to the upper arm in such a way that the resulting force on load 2 in minimised.

Prerequisites for tool loads

Before running the LoadIdentify service routine for a tool load, make sure that:

- the tool is selected in the jogging menu
- the tool is correctly mounted
- axis 6 is close to horizontal and that axis 4 is not too far away from 0 degrees
- the upper arm load is known, if the mass is to be identified
- the axes 3, 5, and 6 are not close to their corresponding working range limits
- the speed is set to 100%
- the system is in manual mode.

Note that LoadIdentify cannot be used for tool0.

Prerequisites for payloads

Before running the LoadIdentify service routine for a payload, make sure that:

- the tool and payload are correctly mounted
- axis 6 is close to horizontal and that axis 4 is not too far away from 0 degrees
- the tool load is known (run LoadIdentify for the tool first)
- the upper arm load is known, if the mass is to be identified
- when using a moving TCP, the tool must be calibrated (TCP)
- when using a stationary TCP, the corresponding work object must be calibrated (user frame and object frame)
- the axes 3, 5, and 6 are not close to their corresponding working range limits
- the speed is set to 100%
- the system is in manual mode.

Running LoadIdentify

To start the load identification service routine you must have an active program in manual mode and the tool and payload that you want to identify must be defined and active in the jogging window.

Step	Action	Info
1.	Start LoadIdentify from the program editor. Press the enabling device and then the Start button on the FlexPendant.	How to start service routines is described in *Running a service routine on page 209*.
2.	Tap **Tool** or **Payload**.	

Continues on next page

Step	Action	Info
3.	Tap **OK** to confirm that the correct tool and/or payload is active in the jogging menu and that the tool load/ payload is correctly mounted.	If it is not correct, release the enabling device and select the correct tool/ payload in the jogging menu. Then return to LoadIdentify, press the enabling device, and press Start. Tap **Retry** and confirm that the new tool/ payload is correct.
4.	When identifying tool loads, confirm that the tool is active. When identifying payloads, confirm that the payload's tool is active and calibrated.	See above.
5.	When identifying payloads with stationary TCP, confirm that the correct work object is active and (preferably) calibrated. If it is correct, tap **OK** to confirm.	See above.
6.	Select identification method. If you select the method where the mass is assumed to be known, remember that the tool/payload that you use must have the correct mass defined. Tap **OK** to confirm.	
7.	Select configuration angle. The optimum is +90 or -90 degrees. If this is impossible, tap **Other** and set the angle. The minimum is plus or minus 30 degrees.	
8.	If the robot is not in a correct position for load identification, you will be asked to jog one or more axes roughly to a specified position. When you have done this tap **OK** to confirm. If the robot is still not in a correct position for load identification, the robot will slowly move to the correct position. Press **Move** to start the movement.	
9.	The robot can go through the load identification movements slowly before performing the load identification. Tap **Yes** if you want a slow test and **No** to proceed to the identification.	This is useful for ensuring that the robot will not hit anything during the identification. However, this will take a lot longer time.
10.	The setup for load identification is now complete. To start the motion, switch to Automatic mode and Motors On. Then tap **Move** to start the load identification movements.	
11.	When the identification is finished, switch back to manual mode, press the enabling device and the Start button. Tap **OK** to confirm.	
12.	The result of the load identification is now presented on the FlexPendant. Tap **Yes** if you want to update the selected tool or payload with the identified parameters or **No** otherwise.	

Continues on next page

Error handling

If the enabling device is released during the load identification (before the movements start), the routine can always be restarted by pressing the enabling device again and then pressing the Start button.

If an error should occur during the load identification movements, the routine must be restarted from the beginning. This is done automatically by pressing Start after confirming the error. To interrupt and leave the load identification procedure, tap **Cancel Call Routine** in the program editor's debug menu.

Limitations for LoadIdentify

Only tool loads and payloads can be identified with LoadIdentify. Thus arm loads cannot be identified using this procedure.

If the load identification movements are interrupted by any kind of stop (program stop, emergency stop, etc.), the load identification must be restarted from the beginning. This is done automatically if you press Start after confirming the error.

If the robot is stopped on a path with program stop and load identification is performed at the stop point, the path will be cleared. This means that no regain movement will be performed to return the robot back to the path.

If the measurement accuracy is lower than 80%, the result of the load identification may have significant errors. In this case, a higher accuracy may be achieved by repeating the LoadIdentify routine. If repeating the routine does not give a higher accuracy, then the torques measured in the identification are probably too small and the tool and/or payload data must be set manually. This is typically the case if the mass of the load is small (10% or less of the maximum load). It can also happen if the load has a particular symmetry property, for instance if the tool load is symmetrical around axis 6. However, even if the measurement accuracy is low some of the identified data may still be correct.

Load identification for suspended robots is only available for bending backward robots, e.g. IRB 140.

Related information

It is also possible to include LoadIdentify in a program by using RAPID instructions. These are described in *RAPID reference manual - Instructions* and *RAPID reference manual - Functions and data types*.

How to enter the data manually is described in *Editing the tool data on page 148*, and *Editing the payload data on page 162*.

The product manual for the robot contain information on how and where to mount the loads.

Load identification for positioners is done with the service routine ManLoadIdentify. This is described in the manual *System settings* for the positioner.

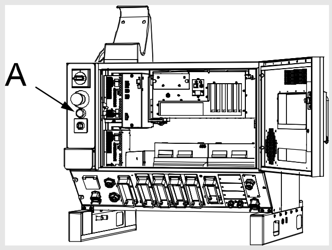
7 Running in production

7.1. Starting programs

Starting programs

This procedure details how to start a program for the first time or to continue running a program that has been stopped.

If your robot system has the multitasking option installed, also see *Using multitasking programs on page 222*.

Step	Action	Info/illustration
1.	Check that all necessary preparations are done to the robot and in the robot cell and that no obstacles exist within the robot's working area.	
2.	Make sure no personnel are inside the robot cell.	
3.	Select operating mode on the control module.	
4.	Press the Motors on button on the control module to activate the robot.	 en0400000783 • A: Button for "Motors on"
5.	Is a program loaded? If yes, proceed to the next step. If no, load a program.	How to load programs is described in section *Handling of programs on page 165*.
6.	Press the Start button on the FlexPendant to start the program.	The button is shown in section *Hardware buttons on page 39*.
7.	Is the **Return to path** dialog box displayed? If yes, return the robot to the path using a suitable method. If no, proceed.	Returning the robot to the path is described in section *Returning the robot to the path on page 224*.

Continues on next page

Continue running after the program is changed

You may always continue a program even if it has been changed.

In automatic mode, a warning dialog may appear to avoid restarting the program if the consequences are unknown.

If you...	then tap...
are sure the changes you have made are not in conflict with the robot’s current position and that the program can continue without danger to equipment or personnel	Yes
are unsure of the consequences your changes might have and want to investigate further	No

Restart from the beginning

This procedure details how to restart a program from the beginning.

Step	Action
1.	On the **ABB** menu, tap **Program Editor**.
2.	Tap **Debug**.
3.	Tap **Move PP to Main.**
4.	Start the program by pressing the Start button on the FlexPendant.

Limitations

Only one program at a time may run, unless your system has the multitasking option. If so several programs may run simultaneously.

If the robot system encounters program code errors while the program is running, it will stop the program and the error is logged in the event log.

7.2. Stopping programs

Stopping programs

If your robot system has the multitasking option installed, see *Using multitasking programs on page 222*.

Step	Action
1.	Check that the ongoing operation is in such a state that it can be interrupted.
2.	Make sure it is safe to stop the program.
3.	Press the **Stop** button on the FlexPendant hardware button set. The button is shown in section *Hardware buttons on page 39*.

DANGER!

Do not use the Stop button in an emergency. Use the emergency stop button.

Stopping a program with the stop button does not mean that the robot will stop moving immediately.

Stopping execution when using hold-to-run or step-by-step execution

When using hold-to-run or step-by-step execution, execution can be stopped according to the following.

Mode	Action	Info
Operation *with* hold-to-run	Release the hold-to-run button	The hold-to-run button is described in section *What is a FlexPendant? on page 38*.
Operation *without* hold-to-run	Press the STOP button	The STOP button is described in section *What is a FlexPendant? on page 38*.
Step-by-step mode	The robot will stop after executing each instruction. Execute the next instruction by pressing the Forward button again.	The STOP and Forward button are described in section *What is a FlexPendant? on page 38*. If you press the STOP button while executing a move instruction, the robot will stop without completing the move.

7.3. Using multitasking programs

Overview

In a multitasking system you may have one or several programs running parallel, for instance in a multimove cell with more than one robot where each robot has its own task and program.

For general information on program handling, see *Handling of programs on page 165*. Multitasking is described in *Application manual - Engineering tools*.

TIP!

Need to know more about tasks and programs? These concepts are described in *The structure of a RAPID application on page 134*.

Manually set up tasks

Tasks need to be set up in order to run as planned. Normally, all tasks are set up on delivery. Setting up tasks is done by defining system parameters of the type Controller. See section *Configuring system parameters on page 270* on how to configure system parameters, or *Technical reference manual - System parameters* for information about the parameters.

You need detailed information to set up tasks manually. Please read your plant or cell documentation for details.

How tasks are run

Tasks may be defined as Normal, Static, or Semistatic. Static and Semistatic tasks are automatically started as soon as a program is loaded into that task.

Normal tasks are started when you press the Start button of the FlexPendant, and stopped when you press the Stop button.

The concepts of Static, Semistatic and Normal are described in *Technical reference manual - System parameters*.

Load, run, and stop multitasking programs

This section describes how to load, run, and stop multitasking programs.

Step	Action
1.	Make sure there is more than one task set up. This is done using system parameters, see section *Using multitasking programs on page 222*.
2.	Load programs to respective task using the Program Editor, this is described in section *Loading an existing program on page 166*.
3.	If one or more task should be disabled, go to the Quickset menu to do this. See section *Quickset menu, Run Mode on page 201*. Deselecting tasks can only be done in manual mode. When switching to automatic mode, an alert box will appear warning that not all tasks are selected to run.
4.	Start program execution by pressing the start button. All active tasks are started.
5.	Stop program execution by pressing the stop button. All active tasks are stopped.

Continues on next page

How to load a program to a task

This section describes how to load a program to a task in a multitasking system. It is assumed that the tasks have been configured.

Step	Action
1.	On the **ABB**menu, tap **Program Editor**.
2.	Tap **Tasks and Programs**.
3.	Tap the task into which you want to load a program.
4.	On the **File** menu, tap **Load Program...**. If you want to open a program in another folder, locate and open that folder. See description in *FlexPendant Explorer on page 75*. The file dialog box appears.
5.	Tap the program you want to load followed by **OK**.
6.	Tap **Close** to close the Program Editor.

Viewing multitasking programs

In the Production Window, there is one tab for each task. To switch between viewing the different tasks, tap on the tabs.

To edit several tasks in parallel, open one Program Editor for each task.

7.4. Returning the robot to the path

About paths and return regions

While a program runs, the robot or additional axis is considered to be *on path* which means that it follows the desired sequence of positions.

If you stop the program the robot is still on path, unless you change its position. It is then considered to be *off path*. However, if the robot is stopped by an emergency or safety stop it may be off path.

If the stopped robot is within the *path return region* you can start the program again, and the robot will return to the path and continue the program.

Note that there is no way to predict the exact return movement for the robot.

TIP!

The path return region is set with system parameters. This is described in the *Technical reference manual - System parameters*.

Returning to path

Step	Action
1.	Make sure there are no obstacles blocking the way and that payloads or work objects are properly placed.
2.	If necessary, put the system in automatic mode and press the Motors on button on the controller to activate the robot's motors.
3.	Press the Start button on the FlexPendant to continue the program from where it stopped. One of two thing will happen: • The robot or axis slowly returns to the path and the program continues. • The return to path dialog is displayed.
4.	If the return to path dialog is displayed, select the proper action.

Select action

If you...	then tap...
want to return to the path and continue the program	**Path**
want to return to the next target position and continue the program	**Position**
don't want to return to the path and continue the program	**Cancel**

7.5 Operating modes

7.5.1. Present operating mode

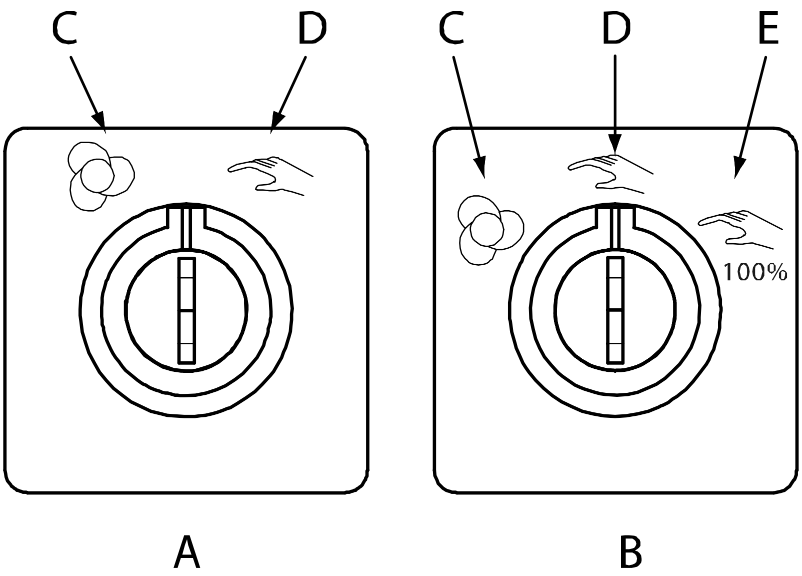
Overview

Check the position of the controller's mode switch or the status bar of the FlexPendant.

Operational mode changes are also logged in the event log.

The mode switch

The mode switch should be in the position as illustrated:



xx0300000466

A	Two position mode switch
B	Three position mode switch
C	Automatic mode
D	Manual reduced speed mode
E	Manual full speed mode

Step	Action	Info
1.	To switch from manual to automatic mode	detailed in *Switching from manual to automatic mode on page 234*.
2.	To switch from automatic to manual mode	detailed in *Switching from automatic to manual mode on page 235*.

Continues on next page

Viewing present mode on the FlexPendant

On the FlexPendant, you can view the present operating mode in the status bar. An example of the status bar is shown below:

en0300000490

A	Operator window
B	Operating mode
C	Active system
D	Controller state
E	Program state
F	Mechanical units, active is highlighted

7.5.2. About the automatic mode

What is the automatic mode?

In automatic mode the enabling device is disconnected so that the robot can move without human intervention.

A robot system in production normally runs in the automatic mode. This mode enables controlling the robot system remotely, for instance by using the controller's I/O signals. An input signal may be used to start and stop a RAPID program, another to activate the robot's motors.

There are also additional safeguarding mechanisms active in automatic mode, not used in manual mode, to increase safety.

Tasks you normally perform in the automatic mode

In automatic mode you normally:

- start and stop processes.
- load, start and stop RAPID programs.
- return the robot to its path when you return to operation after an emergency stop.
- backup the system.
- restore backups.
- tune paths.
- clean tools.
- prepare or replace work objects.
- perform other process oriented tasks.

A well designed system allows you to perform tasks safely and without affecting the running process. In such a system you can at any time enter safeguarded space temporarily having the process stopped by safeguarding mechanisms while you perform the tasks necessary. When you leave safeguarded space the process is resumed.

Please consult your plant or cell documentation for details on process oriented tasks.

CAUTION!

If the robot system is under remote control actions such as starting or stopping process applications and RAPID programs may be overridden. Path tuning may also be disturbed.

In such case perform the mentioned tasks in manual mode.

Limitations in automatic mode

Jogging is not possible in automatic mode. There may also be other specific tasks that you should perform in manual mode to make sure only you are in control of the robot and its movements.

Please consult your plant or system documentation to find out which specific tasks should not be performed in manual mode.

7.5.3. About the manual mode

What is the manual mode?

In manual mode the robot can only move in a reduced and safe speed, and only under manual control.

You need to press the enabling device to activate the robot's motors. The manual mode is most often used when creating programs and when commissioning a robot system.

In some robot systems, there are two manual modes, the normal manual mode, sometimes referred to as Manual Reduced Speed Mode, and then there is a Manual Full Speed Mode.

What is the manual full speed mode?

In manual full speed mode the robot can move in programmed speed but only under manual control.

You need to press the enabling device and the hold-to-run button to activate the robot's motors. The manual full speed mode is most often used when testing programs and commissioning a robot system.

Note that the manual full speed mode is not available in all robot systems.

Tasks you normally perform in manual mode

In manual mode you normally:

- jog the robot back on its path when you return to operation after an emergency stop.
- correct the value of I/O signals after error conditions.
- create and edit RAPID programs.
- tune programmed positions.

Safety in manual mode

When in manual mode some safeguarding mechanisms are disabled since the robot in this mode often is operated with personnel in close proximity. Maneuvering an industrial robot is potentially dangerous and therefore maneuvers should be performed in a controlled fashion, in manual mode the robot is operated in reduced speed, normally 250 mm/s.

7.5.4. Start up in automatic mode

DANGER!

When started the robot may move without warning.

Make sure no personnel are in safeguarded space before you turn on power.

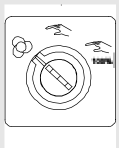
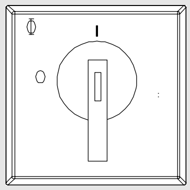
When should I start up in automatic mode?

Start the robot system in automatic mode to resume or start a process or program automatically. Use manual mode for a robot system not yet taken in production or for any other task you need to perform that requires manual mode.

The exact procedure may differ from system to system depending on customizations and programs specifically created for you.

Please consult your plant or cell documentation for details on how to start your specific robot system.

Start up in automatic mode

Step	Action	Illustration
1.	Set the mode switch in the automatic position.	 en0400000794
2.	Turn on the mains power by setting the switch in the on position.	 en0400000793
3.	Did the system start up without errors? If yes, then the procedure is completed. If no, abort.	
4.	After being started, the system will normally be in a safe standby state with motors off awaiting further actions.	

Exceptions

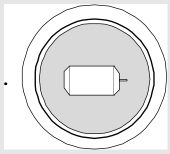
In automatic mode it is possible to start a RAPID program and turn motors on remotely. This means that the system will never enter a safe standby state and the robot may move at any time.

Please consult your plant or cell documentation for details on how your system is configured.

Continues on next page

Manually resume process

Follow this procedure to manually resume the program when the system is not configured for remote restart.

Step	Action	Illustration
1.	On the controller, press the Motors On button to activate the robots motors.	 en0400000795
2.	On the FlexPendant, press the Start button to start the program.	
3.	Did the program start without errors? If yes, the procedure is completed. If no, consult your plant or cell documentation for fault tracing guidelines.	Error handling is detailed in *Trouble shooting manual - IRC5*.

Limitations

It may not always be desired to resume or start a program. The work piece currently in process should perhaps be discarded, or an ongoing glue or weld should perhaps not be continued.

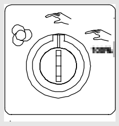
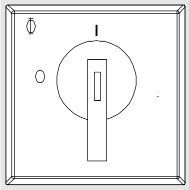
Please consult your plant or cell documentation for guidelines.

7.5.5. Start up in manual mode

When should I start the system in manual mode?

Start the system in manual mode when there is no process or program to be resumed or started or when you need to perform operations not possible in automatic mode such as program editing and jogging.

Start up in manual mode

Step	Action	Illustration
1.	Set the mode switch in the manual mode position.	 en0400000807
2.	Turn on the main power by setting the switch in the on position.	 en0400000793
3.	Did the system start up without errors? If yes, then the procedure is completed. If no, please consult your plant or cell documentation for fault tracing guidelines.	Error handling is detailed in *Trouble shooting manual - IRC5*.
4.	After being started, the system will be in a safe standby state awaiting further actions.	

Where do I go from here?

If you want to...	...then read...
create RAPID programs	*Handling of programs on page 165*.
jog the robot	*Introduction to jogging on page 105*.
work with or change tools, work objects or payloads	*Selecting tool, work object, and payload on page 112*.

7.5.6. Running programs in automatic mode

Running programs in automatic mode

This section details how to run programs in automatic mode.

Step	Action	Info
1.	Switch the robot to Automatic Mode.	How to do this is detailed in section *Switching from manual to automatic mode on page 234*
2.	 xx Before running the robot, please observe the safety information in section *DANGER - Moving manipulators are potentially lethal! on page 18*	
3.	Select the program to be started.	How to load a program is detailed in section *Loading an existing program on page 166*.
4.	Choose in what mode to start the program and start.	
5.	Press the Start button on the FlexPendant.	All FlexPendant buttons are shown in section *What is a FlexPendant? on page 38*

7.5.7. Running programs in manual mode

Running programs in manual mode

This section details how to run programs in manual mode.

Step	Action	Info
1.	Switch the robot to manual mode.	How switch to manual mode is described in section *Switching from automatic to manual mode on page 235*.
2.	 xx **DANGER!** Before running the robot, please observe the safety information in section *DANGER - Moving manipulators are potentially lethal! on page 18*	
3.	Select the program to be started.	How to open a program is detailed in section *Loading an existing program on page 166*.
4.	Select in what run or step mode to start the program.	How to select start mode is detailed in section *How to use the hold-to-run function on page 196*.
5.	Press and hold the enabling device and then press the Start button on the FlexPendant.	All FlexPendant buttons are shown in section *What is a FlexPendant? on page 38*

7.5.8. Switching from manual to automatic mode

When should I put the system in automatic mode?

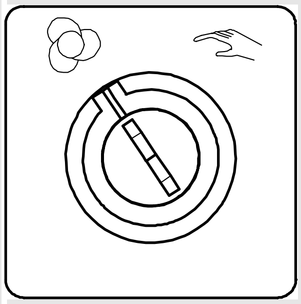
Put the system in automatic mode when you have a process application or a RAPID program that is ready to be run in production.

DANGER!

When put in automatic mode the robot may move without warning.

Make sure no personnel are in safeguarded space before you change operating mode.

Switching from manual to automatic mode

Step	Action	Illustration
1.	Set the mode switch in the automatic position. A mode change dialog is displayed.	 xx0300000467
2.	Tap **OK** to close the dialog. If you change the switch back to manual mode the dialog will be closed automatically.	
3.	Did the system change mode without errors? If yes, then resume or start the process application or RAPID program. If no, stop and troubleshoot the problem.	How to start programs is described in *Starting programs on page 219*.

NOTE!

If your specific system uses a distributed operators panel, controls and indicators may not be placed exactly as described in this manual. Please consult your plant or cell documentation for details.

Controls and indicators do however look and function the same way.

When can I start using the robot system?

As long as the mode change dialog is displayed programs cannot be started and the robot's motors cannot be activated either manually or remotely.

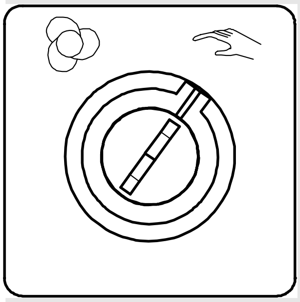
Exceptions

In automatic mode it is possible to start a RAPID program and turn motors on remotely. This means that the system will never enter a safe standby state and the robot may move at any time.

Please consult your plant or cell documentation for details on how your system is configured.

7.5.9. Switching from automatic to manual mode

Switching from automatic to manual mode

Step	Action	Illustration
1.	Set the mode switch in the manual position.	 xx0300000468
2.	Did the system change mode without errors? If yes, then this procedure is completed. If no, try to locate the error.	Error handling is detailed in *Trouble shooting manual - IRC5*.

NOTE!

If your specific system uses a distributed operators panel, controls and indicators may not be placed exactly as described in this manual. Please consult your plant or cell documentation for details.

Controls and indicators do however look and function the same way.

7.5.10. Switching to manual full speed mode

When should I use the manual full speed mode?

Use full speed manual mode when the program is to be tested at full speed.

The manual full speed mode allows you to run the program at full speed while still having access to all the available debugging functions of the program editor.

DANGER!

Testing at full speed is dangerous.

Make sure no personnel are in safeguarded space when starting the program.

Switching to manual full speed mode

Step	Action	Info
1.	Set the mode switch to the manual full speed position.	
2.	Did the system change mode without errors? If yes, then this procedure is completed. If no, try to locate the error.	Error handling is detailed in *Trouble shooting manual - IRC5*.

FlexPendant alert

When changing mode a dialog is displayed on the FlexPendant to alert you about the change of mode. Tap **OK** to close the dialog.

If you change the switch back to the previous mode the dialog will be closed automatically and there will be no change in mode.

8 Handling inputs and outputs, I/O

8.1. Inputs and Outputs, I/O

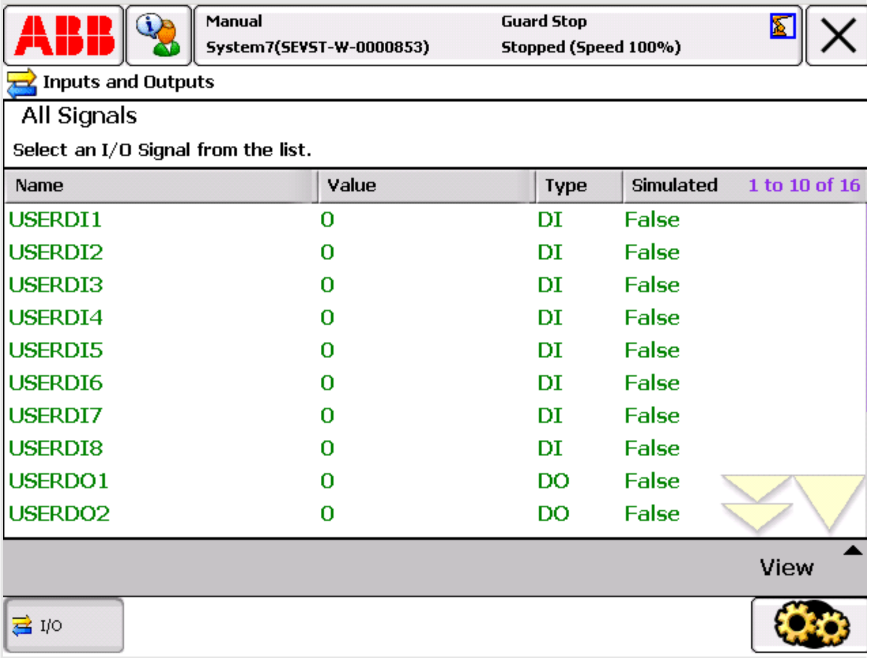
Overview

I/O signal properties is used to view the input and output signals and their values. Signals are configured with system parameters, see section *Configuring system parameters on page 270*.

Viewing signals

This section details how to view a list of all signals.

Step	Action
1.	On the ABB menu tap **Inputs and Outputs**. The list of Most Common I/O signals is displayed.
2.	Tap **View** and then **All signals** to change the selection of signals in the list.



en0400000770

Related information

Simulating and changing signal values on page 238.

Configuring Most Common I/O on page 289.

Configuring system parameters on page 270.

8.2. Simulating and changing signal values

Simulating and changing signal values

A signal can be changed into a simulated signal and the value of the signal can be changed. More information on how to change the signal's properties is described in the section Control Panel, *Configuring Most Common I/O on page 289*.

Step	Action
1.	On the **ABB** menu, tap **I/O**. A list of most common signals is displayed. See section *Configuring Most Common I/O on page 289*.
2.	Tap on a signal.
3.	Tap on **Simulate** to change the signal into a simulated signal. Tap on **Remove Simulation** to remove the simulation from the signal.
4.	Tap on **123...** to change the signal's value. The soft numeric keyboard is displayed. Enter the new value and tap **OK**.

8.3. Viewing signal group

Viewing signal group

This section details how to view signal groups.

Step	Action
1.	On the **ABB** menu, tap **I/O**. A list of most common signals is displayed. See section *Configuring Most Common I/O on page 289*.
2.	In the **View** menu, tap **Groups**.
3.	Tap on the signal group's name in the list and then tap **Properties**. Or tap twice on the signal group's name. The signal group's properties is displayed.

8.4. Safety I/O signals

General

In the controller's basic and standard form, certain I/O signals are dedicated to specific safety functions. These are listed below with a brief description of each.

All signals can be viewed in the I/O menu on the FlexPendant.

Safety I/O signals

The list below contains the safety I/O signals as used by the standard system.

Signal name	Description	Bit value condition	From - To
ES1	Emergency stop, chain 1	1 = Chain closed	From panel board to main computer
ES2	Emergency stop, chain 2	1 = Chain closed	From panel board to main computer
SOFTESI	Soft Emergency stop	1 = Soft stop enabled	From panel board to main computer
EN1	Enabling device1&2, chain 1	1 = Enabled	From panel board to main computer
EN2	Enabling device1&2, chain 2	1 = Enabled	From panel board to main computer
AUTO1	Op mode selector, chain 1	1 = Auto selected	From panel board to main computer
AUTO2	Op mode selector, chain 2	1 = Auto selected	From panel board to main computer
MAN1	Op mode selector, chain 1	1 = MAN selected	From panel board to main computer
MANFS1	Op mode selector, chain 1	1 = Man. full speed selected	From panel board to main computer
MAN2	Op mode selector, chain 2	1 = MAN selected	From panel board to main computer
MANFS2	Op mode selector, chain 2	1 = Man. full speed selected	From panel board to main computer
USERDOOVLD	Over load, user DO	1 = Error, 0 = OK	From panel board to main computer
MONPB	Motors-on pushbutton	1 = Pushbutton pressed	From panel board to main computer
AS1	Auto stop, chain 1	1 = Chain closed	From panel board to main computer
AS2	Auto stop, chain 2	1 = Chain closed	From panel board to main computer
SOFTASI	Soft Auto stop	1 = Soft stop enabled	From panel board to main computer
GS1	General stop, chain 1	1 = Chain closed	From panel board to main computer
GS2	General stop, chain 2	1 = Chain closed	From panel board to main computer

Continues on next page

Signal name	Description	Bit value condition	From - To
SOFTGSI	Soft General stop	1 = Soft stop enabled	From panel board to main computer
SUPES1	Superior stop, chain1	1 = Chain closed	From panel board to main computer
SUPES2	Superior stop, chain2	1 = Chain closed	From panel board to main computer
SOFTSSI	Soft Superior stop	1 = Soft stop enabled	From panel board to main computer
CH1	All switches in run chain 1 closed	1 = Chain closed	From panel board to main computer
CH2	All switches in run chain 2 closed	1 = Chain closed	From panel board to main computer
ENABLE1	Enable from MC (read back)	1 = Enable, 0 = break chain 1	From panel board to main computer
ENABLE2_1	Enable from AXC1	1 = Enable, 0 = break chain 2	From panel board to main computer
ENABLE2_2	Enable from AXC2	1 = Enable, 0 = break chain 2	From panel board to main computer
ENABLE2_3	Enable from AXC3	1 = Enable, 0 = break chain 2	From panel board to main computer
ENABLE2_4	Enable from AXC4	1 = Enable, 0 = break chain 2	From panel board to main computer
PANFAN	Superv. of fan in Control module	1 = OK, 0 = Error	From panel board to main computer
PANEL24OVLD	Overload, panel 24V	1 = Error, 0 = OK	From panel board to main computer
DRVOVLD	Overload, drive modules	1 = Error, 0 = OK	From panel board to main computer
DRV1LIM1	Read back of chain 1 after limit switches	1 = Chain 1 closed	From axis computer to main computer
DRV1LIM2	Read back of chain 2 after limit switches	1 = Chain 2 closed	From axis computer to main computer
DRV1K1	Read back of contactor K1, chain 1	1 = K1 closed	From axis computer to main computer
DRV1K2	Read back of contactor K2, chain 2	1 = K2 closed	From axis computer to main computer
DRV1EXTCONT	External contactors closed	1 = Contactors closed	From axis computer to main computer
DRV1PANCH1	Drive voltage for contactor-coil 1	1 = Voltage applied	From axis computer to main computer
DRV1PANCH2	Drive voltage for contactor-coil 2	1 = Voltage applied	From axis computer to main computer
DRV1SPEED	Read back of op. mode selected	0 = Man. mode low speed	From axis computer to main computer
DRV1TEST1	A dip in run chain 1 has been detected	Toggled	From axis computer to main computer
DRV1TEST2	A dip in run chain 2 has been detected	Toggled	From axis computer to main computer

Continues on next page

Signal name	Description	Bit value condition	From - To
SOFTESO	Soft Emergency stop	1 = Set soft E-stop	From main computer to panel board
SOFTASO	Soft Auto stop	1 = Set soft Auto stop	From main computer to panel board
SOFTGSO	Soft General stop	1 = Set soft General stop	From main computer to panel board
SOFTSSO	Soft Superior stop	1 = Set soft Sup. E-stop	From main computer to panel board
MOTLMP	Motors-on lamp	1 = Lamp on	From main computer to panel board
ENABLE1	Enable1 from MC	1 = Enable, 0 = break chain 1	From main computer to panel board
TESTEN1	Test of Enable1	1 = Start test	From main computer to panel board
DRV1CHAIN 1	Signal to interlocking circuit	1 = Close chain 1	From main computer to axis computer 1
DRV1CHAIN 2	Signal to interlocking circuit	1 = Close chain 2	From main computer to axis computer 1
DRV1BRAKE	Signal to brake-release coil	1 = Release brake	From main computer to axis computer 1

9 Handling the event log

9.1. Accessing the event log

Event log

Open the event log to:

- view all present entries.
- study specific entries in detail.
- handle the log entries, such as saving or deleting.

The log can be printed by using RobotStudio^{Online}.

Open and close the event log

This section details how to open the event log.

Step	Action
1.	Tap the status bar. The status window is displayed.
2.	Tap **Event Log**. The event log is displayed.
3.	If the log contents do not fit into a single screen, it can be scrolled.
4.	Tap a log entry to view the event message.
5.	Tap the status bar again to close the log.

9.2. Deleting log entries

Why should I delete log entries?

Logs can be deleted to increase available disk space. Deleting log entries is often a good way to trace faults since you remove old and insignificant log entries not related to the problem you are trying to solve.

Delete all log entries

Step	Action
1.	Tap the status bar, then the **Event Log** tab to open the event log.
2.	On the **View**menu, tap **Common**.
3.	Tap **Delete** and then **Delete all logs**. A confirmation dialog is displayed.
4.	Tap **Yes** to delete, or **No** to keep the log intact.

Delete log entries of a specific category

Step	Action
1.	Tap the status bar, then the **Event Log** tab to open the event log.
2.	On the **View** menu, tap the category of choice.
3.	Tap **Delete** and then **Delete log**. A confirmation dialog is displayed.
4.	Tap **Yes** to delete, or **No** to keep the log intact.

9.3. Saving log entries

Why should I save log entries?

You should save log entries when:

- you need to clear the log but want to keep the current entries to be viewed later.
- you want to send log entries to support to solve a problem.
- you want to keep log entries for future reference.

NOTE!

The log can keep up to 20 entries per category and up to 1000 entries in the all events list. When the buffer is full the oldest entries will be overwritten and lost.

There is no way to retrieve these lost log entries.

Save all log entries

This section details how to save all log entries.

Step	Action
1.	Tap the status bar to open the event log.
2.	Tap **Save all logs as**. The file dialog is displayed.
3.	If you want to save the log in a different folder, locate and open the folder.
4.	In the **File name** box, type a name for the file.
5.	Tap **Save**.

10 Systems

10.1. About systems

Systems

The FlexController can have one or more systems installed. These are created and managed using RobotStudio Online.

On the FlexPendant you can backup, restart, and select systems, and make modifications to configurations using the system parameters.

For more information on how to create or work with systems, please read *Operator's manual - RobotStudio Online*.

10.2 Memory and file handling

10.2.1. What is “the memory”?

Overview

When using the term “memory”, a number of things may be implied:

- The main computer RAM memory
- The system hard disk drive or flash disk drive
- The hard disk of some other unit connected to the same LAN as the robot system, serving as a storage for software.

Main computer RAM memory

The RAM memory is the main computer primary memory located on the computer motherboard. The memory is used by the processor during all program execution.

The contents of the RAM memory during operation is described in section *The structure of the main computer RAM memory contents on page 249*

System hard/flash disk drive

This is the main mass storage unit of the control module, and is located in the front of the Control Module. Depending on controller version, it may be a hard disk drive or a flash disk drive and may vary in size.

It contains all necessary software for operating the robot, and is the unit on which the RobotWare is installed.

When starting up, data is loaded into the RAM memory from the disk drive.

When powering down, the image.bin is saved here. The contents of the image.bin is described in section *The structure of the main computer RAM memory contents on page 249*

LAN unit

This may be used as extra mass storage device if the one in the controlled is not sufficient. It is not normally considered a part of the robot system

10.2.2. The structure of the main computer RAM memory contents

General

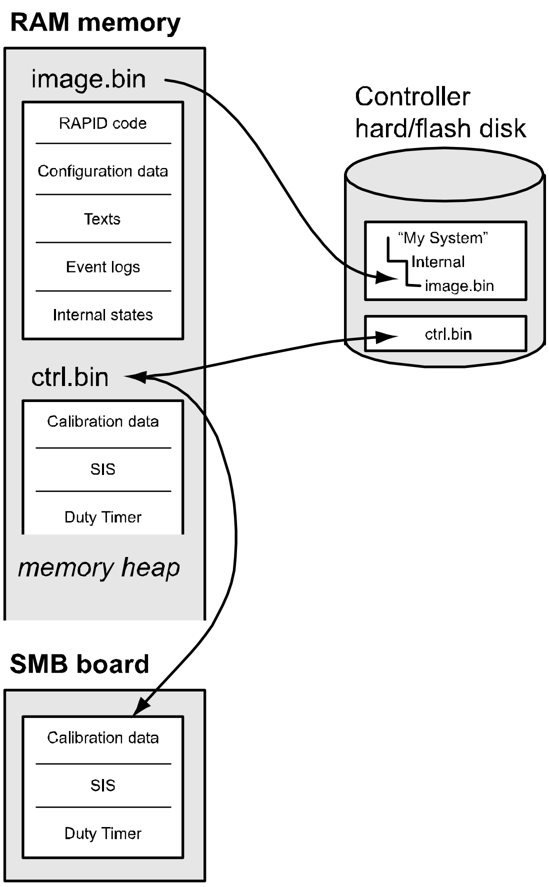
This section describes what the main computer RAM memory contains during normal operation.

The term “RAM memory” means the main computer primary memory, i.e. the memory modules with which the main computer processor works during normal operation.

The generic term “memory” is described in section *What is “the memory”? on page 248*

Illustration of the RAM memory

Each part of the illustration is described in the table below.



en0500001446

Continues on next page

Parts

Part	Function
RAM memory	The main computer memory modules, located on the computer motherboard. The processor reads and writes to this memory during program execution. The size of the RAM memory may vary, but increasing the size *will not improve* computer performance unless a number of hard- and software changes are made to the robot system.
image.bin	When the system is powered OFF, intentionally or due to power failure, the image.bin file is saved to the controller hard/flash disk. It is an internal file, created by the system during operation, usually invisible to the user. When performing a “warm start” of the system, the complete image.bin file is reloaded into the RAM memory. Other types of restarts may start with another system, etc, which is described in the *Operator’s manual - IRC5 with FlexPendant*.
ctrl.bin	This file contains, among other things: • robot identity data • calibration data • SIS data • duty timer data The file is stored on the SMB board on robot delivery. Data may then be transferred to the controller as detailed in the *Operator’s manual - IRC5 with FlexPendant*. NOTE that the ctrl.bin file is *not stored* in the system specific file on the hard/ flash disk drive. This means that all data in the file will be retained even if the system software is updated or in any other way replaced.
SMB board	The SMB board (serial measurement board) is normally fitted on the mechanical unit, and contains among other things, data from the ctrl.bin file. How to handle the data on the SMB board, moving data between SMB and controller, etc is detailed in the *Operator’s manual - IRC5 with FlexPendant*.
Controller hard/flash disk	The main mass storage unit of the control module, located in the Computer Unit. Depending on controller version, it may be a hard disk drive or a flash disk drive and may vary in size. It contains all necessary software for operating the robot, and is the unit on which the RobotWare is installed. When starting up, data is loaded into the RAM memory from the disk drive. When powering down, the image.bin file is automatically saved here.
RAPID code	This section contains all executable RAPID code, whether written by ABB or the customer.
Configuration data	This data is basically the contents of the configuration files: • proc.cfg • moc.cfg • sio.cfg • mmc.cfg • sys.cfg Each file contains the settings made when creating and defining the system, options etc. The configuration files may not be changed after creation, but their contents may be checked as detailed in the Trouble Shooting Manual - IRC5. When changing the contents of the configuration files, ABB strongly recommends using the tool RobotStudio$^{\text{Online}}$ to reduce the risk of introducing errors.

Continues on next page

Part	Function
Texts	Some of the texts used by the system during operation, in all languages selected when creating the system.
Event logs	All events logged in all event logs. This means that the logs will be saved even if a power failure occurs, which in turn, simplifies finding the fault causing the power failure.
Internal states	This is data recording the state and position of all robot axes, all I/O, the state of each manipulator connected to a Multimove system, etc. This data is constantly updated during operation. This enables the system to instantly return to it’s previous state if the system for any reason stops, there is a power failure or the robot collides with an obstacle etc.
Calibration data	This is calibration data for one robot, i.e. all data describing the calibration position for all six axes of one robot.
SIS	This is service data related to the SIS system (Service Interval System). This means that SIS data will be kept by the robot even if it’s controller is replaced.
Duty timer	This is the Duty timer data. This means that duty timer count will be kept by the robot even if it’s controller is replaced.
“My system”	This is the directory in which the RobotWare is stored after installation. The image file is stored in the directory “Internal”. NOTE that the ctrl.bin file is *not stored* here, which means that the contents of the image.bin file will be retained even if updating the system software during operation.

10.2.3. File handling

File handling and storing

Backups, programs, and configurations etc. are saved as files in the robot system. These files are handled either in a specific FlexPendant application, such as the Program editor, or using the FlexPendant Explorer.

Files can be stored on a number of different drives, or memory devices, such as:

- Controller hard disk
- Portable PC
- USB device
- Other network drives

These drives are used the same way and available in the FlexPendant Explorer or when saving or opening files using an application on the FlexPendant.

USB memory information

IRC5 is equipped with a USB port on the controller module, see chapter *Buttons on the controller on page 47*.

A USB memory is normally detected by the system and ready to use within a few seconds from plugging in the hardware. A plugged in USB memory is automatically detected during system start up.

It is possible to plug in and unplug a USB memory while the system is running. However, observing the following precautions will avoid problems:

- Do not unplug a USB memory immediately after plugging in. Wait at least five seconds, or until the memory has been detected by the system.
- Do not unplug a USB memory during file operations, such as saving or copying files. Many USB memories indicates ongoing operations with a flashing LED.
- Do not unplug a USB memory while the system is shutting down. Wait until shutdown is completed.

Please also note the following limitations with USB memories:

- There is no guarantee that all USB memories are supported.
- Some USB memories have a write protection switch. The system is not able to detect if a file operation failed due to the write protection switch.

Related information

For more information on trouble shooting, see *Trouble shooting manual - IRC5*.

What is “the memory”? on page 248.

10.3 Restart procedures

10.3.1. Restart overview

When do I need to restart a running controller?

ABB robot systems are designed to operate unattended for long times. There is no need to periodically restart functioning systems.

Restart the robot system when:

- new hardware has been installed.
- the robot system configuration files have been changed.
- a new system has been added and is to be used.
- a system failure (SYSFAIL) has occurred.

Restart types

A number of restart types are available:

Situation:	Restart type:	Detailed in section:
You want to restart and use the current system. All programs and configurations will be saved.	W-start (Warm restart)	*Restart and use the current system (warm start) on page 257*.
You want to restart and select another system. The Boot application will be launched at startup.	X-start (Xtra restart)	*Restart and select another system (X-start) on page 258*.
You want to switch to another installed system **or** install a new system **and**, at the same time, remove the current system from the controller. **Warning!** *This can not be undone. The system and the RobotWare system package will be deleted.*	C-start (Cold restart)	*Restart and delete the current system (C-start) on page 259*.
You want to delete all user loaded RAPID programs. **Warning!** *This can not be undone.*	P-start	*Restart and delete programs and modules (P-start) on page 260*.
You want to return to the default system settings. **Warning!***This will remove all user defined programs and configurations from memory and restart with default factory settings.*	I-start (Installation restart)	*Restart and return to default settings (I-start) on page 261*.
You want to restart the current system using the system data from most recent successful shut down.	B-start	*Restart from previously stored system data (B-start) on page 262*.
You want to shut down and save the current system and shut down the main computer.	Shutdown	*Shutting down on page 69*.

Related information

More information about the different restart procedures is also described in *Trouble shooting manual - IRC5*.

10.3.2. Using the boot application

Boot application

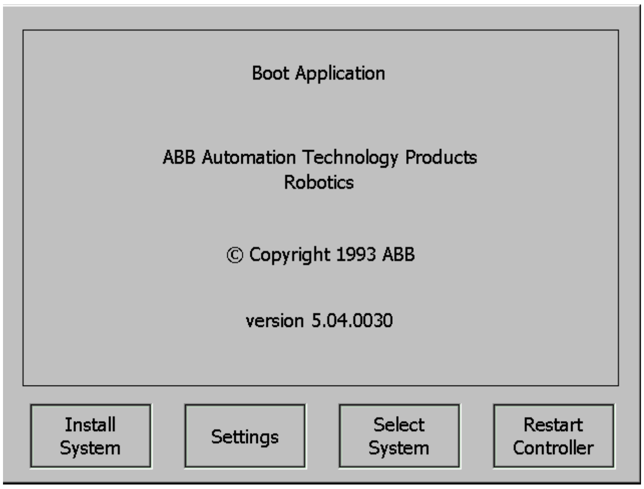
The boot application is primarily used to start up the system when no RobotWare is installed, but may also be used for other purposes, such as changing the system to start. You can also use RobotStudio Online, see *Operator's manual - RobotStudio Online*.

Purpose of the boot application

The boot application is installed in the controller at delivery and may be used to:

- install systems
- set or check network settings
- select a system/switch between systems from the mass storage memory
- load the system from USB memory units or network connections

The illustration shows the boot application main screen. The buttons and functions available are described below.



en0400000894

Installing a system

This procedure may take several minutes.

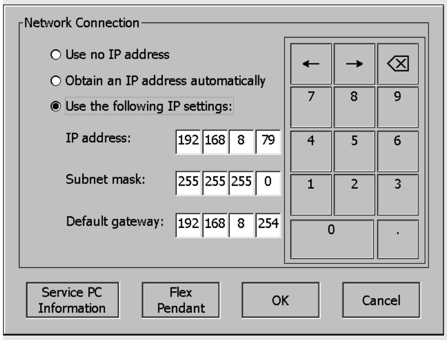
Step	Action	Info
1.	You may reach the boot application by performing an X-start.	How to perform an X-start is detailed in section *Restart and select another system (X-start) on page 258*.
2.	In the boot application, tap **Install System**. A dialog box is displayed urging you to connect a USB memory.	Systems are created using the System Builder in RobotStudio Online. See *Operator's manual - RobotStudio Online*.

Continues on next page

Step	Action	Info
3.	Connect a USB memory containing a system to the computer unit USB port.	How to load a system to the USB memory is detailed in section *Creating boot media* in *Operator's manual - RobotStudio Online*. The USB port is shown in section *Buttons on the controller on page 47*.
4.	Tap **Continue** to proceed. Tap **Cancel** to abort. The system is read from the USB memory, and a dialog box is displayed, urging you to restart.	
5.	Tap **OK**.	The USB memory may be disconnected at this point.
6.	Tap **Restart Controller** and then tap **OK**. The controller is now restarted and the system from the USB memory is installed. The restart may take several minutes.	

Boot application settings

The boot application settings contain IP and network settings.

Step	Action	Info
1.	You may reach the boot application by performing an X-start.	How to perform an X-start is detailed in section*Restart and select another system (X-start) on page 258*.
2.	In the boot application, tap **Settings**.  en0400000902	
3.	Make the appropriate selections: • Use no IP address • Obtain IP address automatically • Use the following settings Use the numerical keyboard to enter the desired values.	How to make these selections is detailed in section *Set up the network connection on page 54*.

Continues on next page

Step	Action	Info
4.	Tap **Service PC information** to display network settings to be used when connection a service PC to the controller's service port.	
5.	Tap **FlexPendant** to display FlexPendant software versions. Tap **Advanced** to display the boot loader version.	

Selecting system

Step	Action	Info
1.	You may reach the boot application by performing an X-start.	How to perform an X-start is detailed in section *Restart and select another system (X-start) on page 258*.
2.	In the boot application, tap **Select System**. A dialog box is displayed showing the available installed systems.	
3.	Tap the system to be selected and then **Select**. The selected system is displayed in the box Selected System.	
4.	Tap **Close**. A dialog box is shown urging you to restart to be able to use the selected system.	

Restarting controller

Step	Action	Info
1.	You may reach the boot application by performing an X-start.	How to perform an X-start is detailed in section *Restart and select another system (X-start) on page 258*.
2.	In the boot application, tap **Restart System**. A dialog box is displayed specifying the selected system.	
3.	Tap **OK** to restart using the selected system or **Cancel** to abort.	

10.3.3. Restart and use the current system (warm start)

What happens with my current system?

The current system will be stopped.

All system parameters and programs will be saved to an image file.

During the restart process the system's state will be resumed. Static and semistatic tasks will be started. Programs can be started from the point they where stopped.

Restarting this way will activate any configuration changes entered using RobotStudio^{Online}.

Restart and use the current system

This section describes how to restart and use the current system.

Step	Action	Info
1.	On the **ABB**menu, tap **Restart**. The restart dialog is displayed.	
2.	Tap **Warm Start** to restart the controller using the current system.	To select another type of start, tap **Advanced**. Detailed information about advanced starts is given in *Restart overview on page 253*.

10.3.4. Restart and select another system (X-start)

What happens with my current system?

The current system will be stopped.

All system parameters and programs will be saved to an image file, so that the system's state can be resumed later.

Restart and select another system

This section describes how to restart and select another system.

Step	Action	Info
1.	Make sure the power to the controller cabinet is switched on.	
2.	On the **ABB**menu, tap **Restart**. The restart dialog is displayed.	
3.	Tap **Advanced...** to select restart method. The select restart method dialog is displayed.	
4.	Tap **X-start**, then tap **OK**. A dialog letting you confirm that you really want to restart is displayed.	
5.	Tap **X-Start** to restart the controller. The controller is restarted. After the startup procedure the boot application is started.	
6.	Use the boot application to select system.	How to use the boot application is detailed in *Using the boot application on page 254*.
7.	Tap **Close**, then **OK** to return to the boot application.	
8.	Tap **Restart** to restart the controller using the selected system.	

10.3.5. Restart and delete the current system (C-start)

What happens with my current system?

Your current system will be stopped.

All contents, backups and programs, in the system directory **will be deleted**. This means it will be **impossible to resume** this system's state in any way. A new system must be installed using RobotStudio^{Online}.

Restart and delete the current system

This section describes how to restart and delete the current system.

Step	Action	Info
1.	On the **ABB**menu, tap **Restart**. The restart dialog is displayed.	
2.	Tap **Advanced...** to select restart method. The select restart method dialog is displayed.	
3.	Tap **C-start**, then tap **OK**. A dialog letting you confirm that you want to restart is displayed.	
4.	Tap **C-start** to restart the controller. A dialog letting you confirm that you want to restart is displayed.	
5.	Perform any of the following procedures: • Select an already installed system and restart. • Install another system from RobotStudio^{Online} or from a USB memory.	How to restart and select another system is described in section *Restart and select another system (X-start) on page 258*. RobotStudio^{Online} is described in *Operator's manual - RobotStudio Online*

10.3.6. Restart and delete programs and modules (P-start)

What happens with my current system?

After restart the system's state will be resumed except for manually loaded programs and modules. Static and semistatic tasks are started from the beginning, not from the state they had when the system was stopped.

Modules will be installed and loaded in accordance with the set configuration. System parameters will not be affected.

Restart and delete programs and modules

This section describes how to restart and delete user loaded programs and modules.

Step	Action
1.	On the **ABB**menu, tap **Restart**. The restart dialog is displayed.
2.	Tap **Advanced...** to select restart method. The select restart method dialog is displayed.
3.	Tap **P-start**, then tap **OK**. A dialog letting you confirm that you really want to restart is displayed.
4.	Tap **P-start** to restart the controller. The controller is restarted using the current system. After the startup procedure no programs or modules are open.

10.3.7. Restart and return to default settings (I-start)

What happens to my current system?

After restart, the system's state will be resumed but any changes done to system parameters and other settings will be lost. Instead, system parameters and other settings are read from the originally installed system on delivery.

For example, this returns the system to the original factory system state.

Restart and return to default settings

This section describes how to restart and return to default settings.

Step	Action
1.	On the **ABB**menu, tap **Restart**. The restart dialog is displayed.
2.	Tap **Advanced...** to select restart method. The select restart method dialog is displayed.
3.	Tap **I-start**, then tap **OK**. A dialog letting you confirm that you really want to restart is displayed.
4.	Tap **I-start** to restart the controller. The controller is restarted using the current system. Changes to system parameters and other settings are lost.

10.3.8. Restart from previously stored system data (B-start)

What happens with my current system?

The current system is in system failure mode since the previous session was shut down without saving the image file correctly. All changes made to the system before the shut down has been lost. Therefore, the system needs to be restarted from the most recent successful shut down or load another system.

Note that all changes made to the system data since the previous session has been lost.

Restart from previously stored system data

This section describes how to restart from previously stored image file.

Step	Action
1.	On the **ABB**menu, tap **Restart**. The restart dialog is displayed.
2.	Tap **Advanced...** to select restart method. The select restart method dialog is displayed.
3.	Tap **B-start**, then tap **OK**.
4.	Tap **B-start** to restart the controller. The controller is restarted using the system data from the most recent successful shut down.

10.3.9. Reflashing firmware and FlexPendant

Overview of reflashing

After replacing hardware units, such as axis computer, buses, etc., or installing newer versions of RobotWare, the system will automatically attempt reflashing the unit in order to maintain hardware/software compatibility.

Reflashing is loading appropriate firmware (hardware specific software) onto a specific unit running this software during operation.

If RobotWare is upgraded on the controller, then the FlexPendant will reflash, i.e. update to the new version, when connected.

Note that performing any such replacements/updates might require running firmware versions not available! To avoid jeopardizing the function of the system, ABB recommends using the same versions as earlier, unless these are unavailable.

The units currently using the reflash function are:

- Contactor interface board
- Drive units
- FlexPendant
- Profibus master
- Axis computer
- Panel board

Reflashing process

The automatic reflashing process, described below, must not be disturbed by switching off the controller while running:

Step	Event	Info
1.	When the system is restarted, the system checks the versions of the firmware used. These are checked against the hardware versions used.	
2.	If the hardware and firmware versions do not match, the system restarts itself automatically while going to a specific *Update Mode*.	During the Update Mode, an attempt is made to download appropriate firmware onto the hardware while a message is very briefly displayed on the FlexPendant.
3.	Was an appropriate firmware version found? If YES, the reflash will proceed. If NO, the system will stop. Proceed as detailed in section Reflashing firmware failed in the *Trouble shooting manual - IRC5*.	In either case, a message is very briefly displayed on the FlexPendant and stored in the event log. The actual reflashing may take a few seconds or up to a few minutes, depending on the hardware to be reflashed.
4.	After performing a successful reflash, the system restarts.	

Continues on next page

Step	Event	Info
5.	Another check is made for any additional hardware/firmware mismatches.	
6.	Was any additional mismatches found? If YES, the process is repeated until none are found. If NO, the process is complete.	

10.4 Backup and restore systems

10.4.1. What is saved on backup?

General

When performing a backup, or restoring a previously made backup, only certain data is dealt with. This section is a specification and description of these.

What *is* saved?

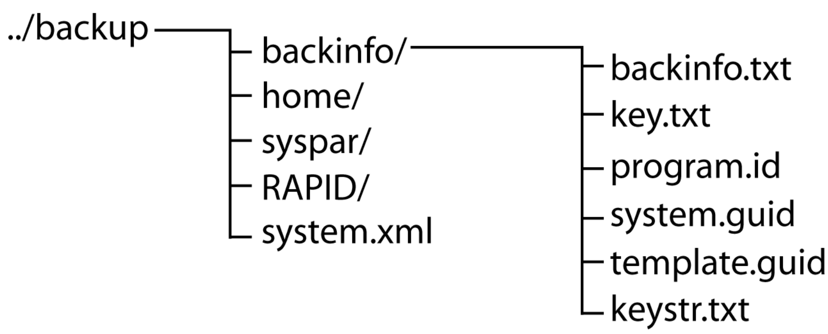
The backup function saves all system parameters, system modules, and program modules in a context.

The data is saved in a directory specified by the user. The directory is divided into four subdirectories, Backinfo, Home, Rapid, and Syspar. System.xml is also saved in the ../backup (root directory) it contains user settings.

Backinfo

consists of the files *backinfo.txt*, *key.id*, *program.id* and *system.guid*, *template.guid*, *keystr.txt*.

The restore part uses *backinfo.txt* when the system is restored. This file must **never** be edited by the user! The files *key.id* and *program.id* may be used to recreate a system, using RobotStudio^Online, with the same options as the backed up system. The *system.guid* is used to identify the unique system the backup was taken from. The *system.guid* and/or *template.guid* is used in the restore to check that the backup is loaded to the correct system. If the *system.guid* and/or *template.guid* do not match, the user will be informed.



en0400000916

Home

a copy of the files in the HOME directory.

Rapid

consists of a subdirectory for each task configured. Every task has one directory for program modules and one for system modules.

The first directory will keep all installed modules. More information on loading modules and programs is given in the *Technical reference manual - System parameters*.

SysPar

contains the configuration files.

Continues on next page

What *is not* saved?

A few things are not saved during backup, and it may be vital to be aware of this, in order to save these separately:

- The environment variable RELEASE: points out the current system pack. System modules loaded with RELEASE: as its path, are not stored in the backup.
- The current value of a PERS object in a installed module is not stored in a backup.

10.4.2. Backup the system

When do I need this?

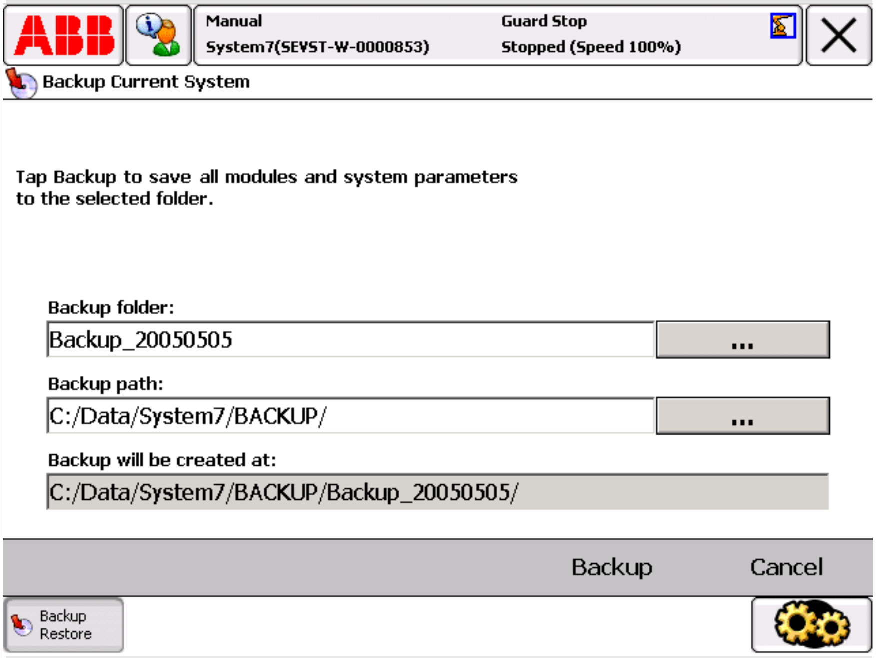
ABB recommends performing a backup:

- *before* installing new RobotWare.
- *before* making any major changes to instructions and/or parameters to make it possible to return to the previous setting.
- *after* making any major changes to instructions and/or parameters and testing the new settings to retain the new successful setting.

Backup the system

This section describes how to backup the system.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>Tap the ABB menu and then tap Backup and Restore.</td></tr>
<tr><td>2.</td><td>Tap Backup Current System.</td></tr>
<tr><td>3.</td><td>Is the displayed backup path the correct one?
If YES: Tap Backup to perform the backup to the selected directory. A backup file named according to the current date is created.
If NO: Tap ... to the right of the backup path and select directory. Then tap Backup. A backup folder named according to the current date is created.



xx0300000441</td></tr>
</table>

10.4.3. Restore the system

When do I need this?

ABB recommends performing a restore:

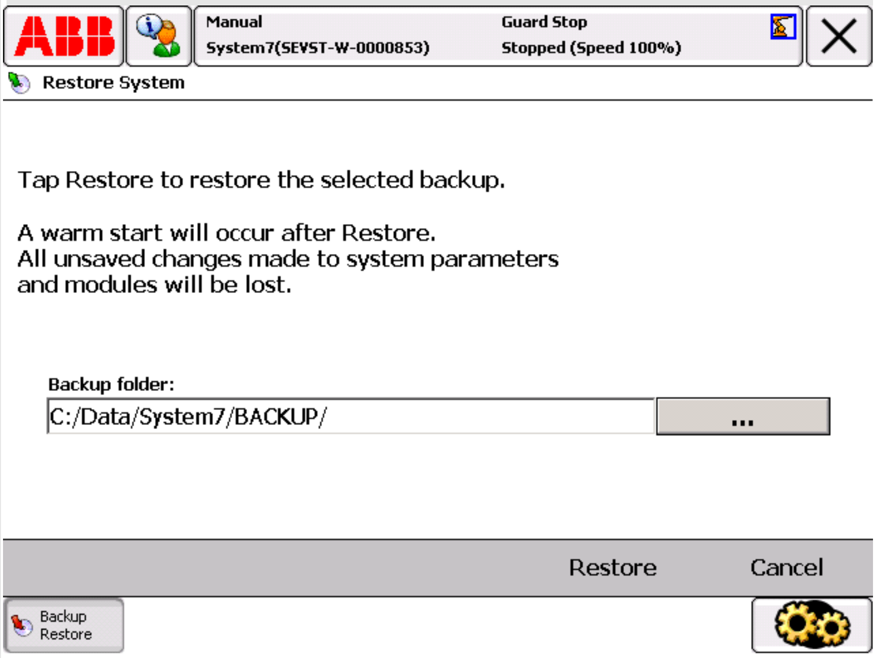
- if you suspect that the program file is corrupt.
- if any changes made to the instructions and/or parameters settings did not prove successful, and you want to return to the previous settings.

During the restore, all system parameters are replaced and all modules from the backup directory are loaded.

The Home directory is copied back to the new system’s HOME directory during the warm start.

Restore the system

This section describes how to restore the system.

Step	Action
1.	On the **ABB** menu, tap **Backup and Restore**.
2.	Tap **Restore System**.
3.	Is the displayed backup folder the correct one? If YES: Tap **Restore** to perform the restore. The restore is performed, and the system is warm started automatically. If NO: Tap **...** to the right of the backup folder and select directory. Then tap **Restore**. The restore is performed, and the system is warm started automatically.  xx0300000442

10.4.4. Important when performing backups!

General

When performing backups or restoring previously made backups, there are several things to keep in mind. Some of these are listed below.

BACKUP directory

A local default backup directory, BACKUP, is automatically created by the system. We recommend using this directory for saving backups! Such backups are not copied to the directory HOME in following backups.

Never change the name of the BACKUP directory.

Also, never change the name of the actual backup to BACKUP, since this would cause interference with this directory.

When is backup possible?

A backup of a system may be performed during program execution. When doing so, a few limitations apply:

- Start program, load program, load module, close program and erase module can not be done during backup in executing state. The RAPID instructions `Load` and `StartLoad` can, however, be used.

What happens during backup?

Beside the obvious, a backup being made, a few thing happen during backup:

- Background tasks continue to execute during a backup.

Duplicated modules?

No save operation is performed in the backup command. This implies that two revisions of the same modules can exist in the backup, one from the program memory saved in Rapid\Task\Progmod\ directory and one from the HOME directory copied to the Backup's Home directory.

Large data amount

Too many files in the HOME directory can result in a very large backup directory. The unnecessary files in the Home directory can then be deleted without any problems.

Fault during backup

If a fault occurs during the backup, e.g. full disk or power failure, the whole backup structure is deleted.

10.5 Configuring systems

10.5.1. Configuring system parameters

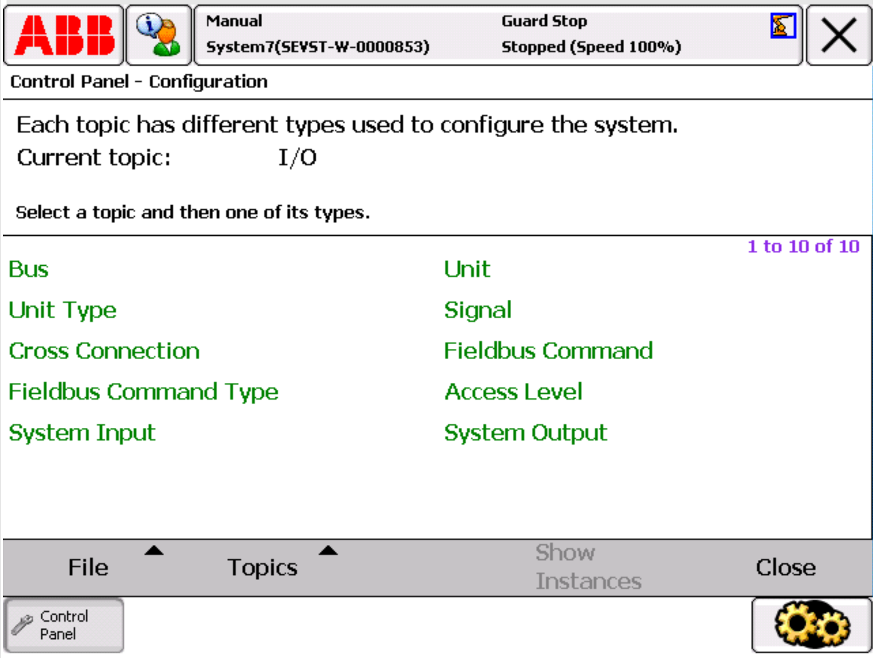
About system parameters

System parameters define the system configuration and are defined to order on delivery. System parameters are edited using the FlexPendant or RobotStudio[Online].

All system parameters are described in *Technical reference manual – System parameters*.

Viewing system parameters

This procedure describes how to view system parameter configurations.

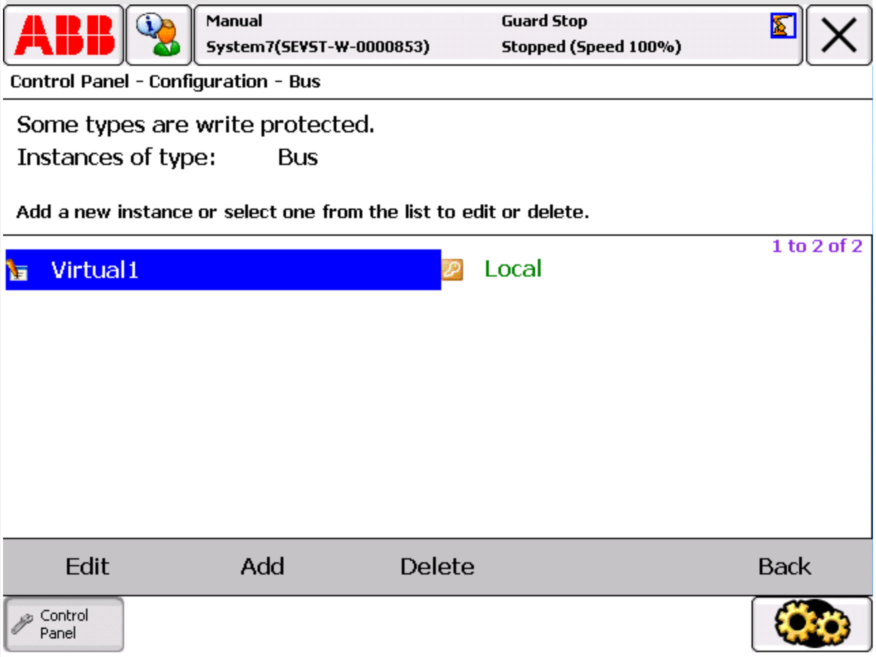
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>On the ABB menu, tap Control Panel.</td></tr>
<tr><td>2.</td><td>Tap Configuration.
A list of available types in the selected topic is displayed.


en0400001149</td></tr>
<tr><td>3.</td><td>Tap Topics to select the topic.
Controller
Communication
I/O
Man-machine Communication
Motion</td></tr>
<tr><td>4.</td><td>Tap File to save, load, or add new parameters from a file. Select folder and save or load. Proceed to section Saving, loading system parameter configurations on page 272.</td></tr>
<tr><td>5.</td><td>Tap to select a type and then tap Show Instances. To edit parameters proceed to section Editing an instance on page 271. To add instances proceed to section Adding a new instance on page 271.</td></tr>
</table>

Continues on next page

Editing an instance

This section describes how to edit an instance of a system parameter type.

Step	Action
1.	In the list of system parameter instances, tap to select an instance and then tap **Edit**. The selected instance is displayed.  en0400001151
2.	Tap a parameter name or its value to edit the value. The way to edit values depend on the data type for the value, e.g. the soft keyboard is displayed for string or numerical values and dropdown menus are displayed for predefined values.
3.	Tap **OK**.

Adding a new instance

This section describes how to add a new instance of a system parameter type.

Step	Action
1.	In the list of system parameter instances, tap **Add**. A new instance with default values is displayed.
2.	Tap the parameter name or its value to edit the value.
3.	Tap **OK**.

Continues on next page

Saving, loading system parameter configurations

This section describes how to save and load system parameter configurations, and how to add parameters from a file.

It is recommended to save the parameter configurations before making larger changes to the robot system. The parameters are saved automatically when performing backups.

Step	Action
1.	In the list of types, tap the **File** menu and tap: • **Load saved parameters** • **Add new parameters** • **Add or replace parameters** • **Save As** to save the selected topic's parameter configurations. • **Save All As** to save all topics' parameter configurations.
2.	Select directory and/or file where you want to save or load the parameters.
3.	Tap **OK**.

11 Calibrating

11.1. How to check if the robot needs calibration

Check robot calibration status

This section describes how to check the robot's calibration status.

Step	Action
1.	On the **ABB** menu, tap **Calibration**.
2.	In the list of mechanical units, check the calibration status.

What kind of calibration is needed?

If the calibration status is...	then...
Not calibrated	the robot must be calibrated by a qualified service technician. See section *Loading calibration data using the FlexPendant on page 276*.
Rev. counter update needed	You must update the revolution counters. How to update the revolution counters is described in section *Updating revolution counters on page 274*.
Calibrated	No calibration is needed.

DANGER!

Do not attempt to perform the fine calibration procedure without the proper training and tools. Doing so may result in incorrect positioning that may cause injuries and property damage.

11.2. Updating revolution counters

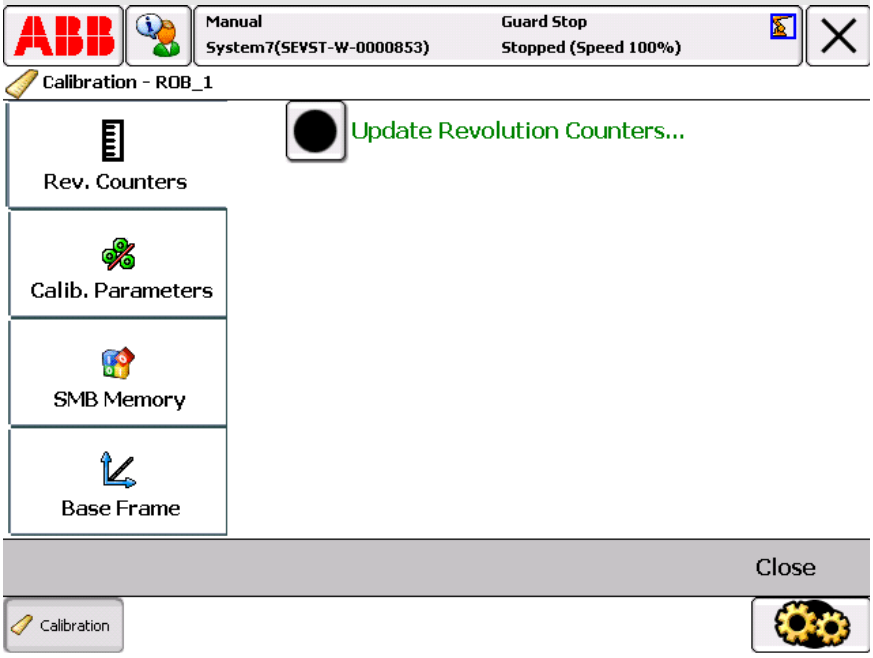
Overview

This section details how to perform a rough calibration of each robot axis, i.e. updating the revolution counter value for each axis, using the FlexPendant. Detailed information about revolution counters and how to update them, with calibration positions and scales, can be found in the respective robot's product manual. Also, see the manuals *Instructions for Levelmeter Calibration* and *Calibration Pendulum Instruction* for information on calibration.

For robots using the Absolute Accuracy option, the calibration data file absacc.cfg must be loaded first.

Storing the revolution counter setting

This procedure details the second step when updating the revolution counter; storing the revolution counter setting.

Step	Action
1.	On the **ABB** menu, tap **Calibration**. All mechanical units connected to the system are shown along with their calibration status.
2.	Tap the mechanical unit in question. A screen is displayed: tap **Rev. Counters**.  en0400000771
3.	Tap **Update revolution counters...**. A dialog box is displayed, warning that updating the revolution counters may change programmed robot positions: • Tap **Yes** to update the revolution counters. • Tap **No** to cancel updating the revolution counters. Tapping **Yes** displays the axis selection window.

Continues on next page

Step	Action
4.	Select the axis to have its revolution counter updated by: • Ticking in the box to the left • Tapping **Select all** to update all axes. Then tap **Update**.
5.	A dialog box is displayed, warning that the updating operation cannot be undone: • Tap **Update** to proceed with updating the revolution counters. • Tap **Cancel** to cancel updating the revolution counters. Tapping **Update** updates the selected revolution counters and removes the tick from the list of axes.
6.	 - **Caution!** If a revolution counter is incorrectly updated, it will cause incorrect robot positioning, which in turn may cause damage or injury! Check the calibration position very carefully after each update. See section *Checking the calibration position* in either of the manuals: *Instructions for Levelmeter Calibration* or *Calibration Pendulum Instruction*, depending on which calibration method to be used. The Product manual for the robot also contains more information about calibration.

11.3. Loading calibration data using the FlexPendant

Overview

This section describes how to load calibration data for using the FlexPendant.

The calibration data is delivered on a diskette and will have to be moved to a USB memory or transferred to the controller through FTP.

Load calibration data

This section describes how to load the calibration data.

Step	Action	Info
1.	On the **ABB** menu, tap **Calibration** and select a mechanical unit. Then tap **Calib. parameters**.	
2.	Tap **Load motor calibration...**. A dialog box is displayed, warning that loading new calibration offset values may change programmed robot positions: • Tap **Yes** to proceed. • Tap **No** to cancel.	Tapping **Yes** results in displaying a file selection window.
3.	For systems *not* running the Absolute Accuracy option the calibration data is normally stored on the serial measurement board (SMB).	In such case, Update data to controller from SMB memory as detailed in section *Serial Measurement Board memory on page 280*
4.	For system running the Absolute Accuracy option, the calibration data is normally delivered on a diskette.	In such case, proceed below.
5.	Select the *file containing the Absolute Accuracy calibration data* to be loaded into the system and tap **OK**. If a file containing invalid calibration data is selected, a dialog box will be displayed. Then re-select a file containing valid calibration data.	*Absacc.cfg* for systems *with* absolute accuracy measurement system

11.4. Editing motor calibration offset

Editing motor calibration offset

This procedure should be used when no specific file with motor calibration data is available, but only the numerical values. These values are normally found on a sticker on the rear of the robot.

Entering motor calibration values may be done in one of three ways:

- From a disk, using the FlexPendant (as detailed in section *Loading calibration data using the FlexPendant on page 276*).
- From a disk, using RobotStudio Online (as detailed in section *Loading calibration data* in *Operator's manual - RobotStudio Online*).
- Manually entering the values, using the FlexPendant (as detailed in section *Editing motor calibration offset on page 277*).

Step	Action	Info
1.	On the **ABB** menu, tap **Calibration**.	
2.	Tap to select mechanical unit and then tap **Calibration Parameters**.	
3.	Tap **Edit motor calibration offset...**. A dialog box is displayed, warning that updating the revolution counters may change programmed robot positions: • Tap **Yes** to proceed. • Tap **No** to cancel. Tapping **Yes** results in displaying a file selection view.	
4.	Tap the axis to have its motor calibration offset edited. The offset value box is opened for that particular axis.	
5.	Use the numerical keyboard to enter the desired value and then tap **OK**. After entering new offset values, a dialog box is displayed, urging you to restart the system to make use of the new values. Perform a warm restart if required.	
6.	After restarting, the contents of the calibration data in the controller cabinet and on the serial measurement board will differ. Update the calibration data.	Detailed in section *Serial Measurement Board memory on page 280*
7.	Update the revolution counters.	Detailed in section *Updating revolution counters on page 274*

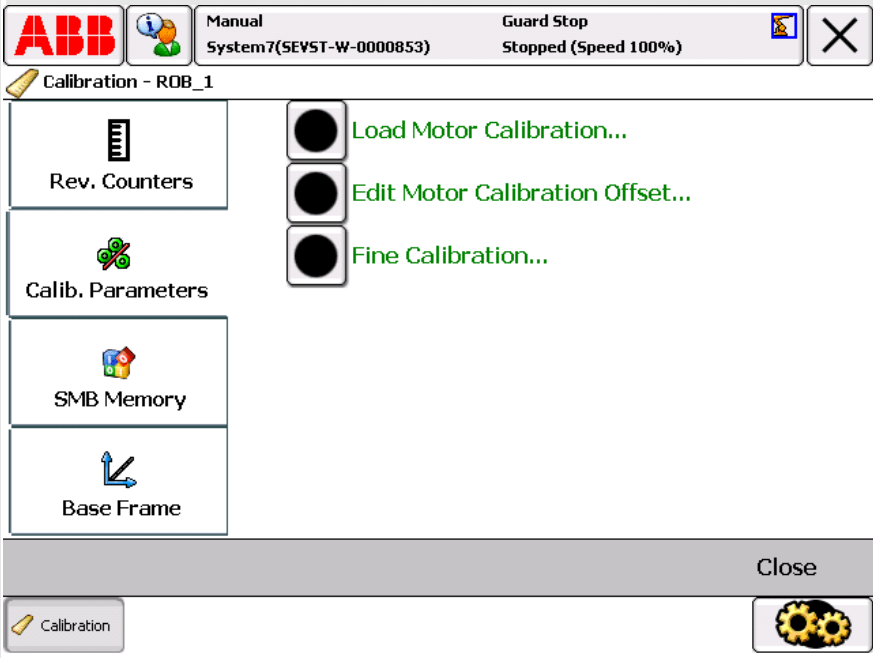
11.5. Fine calibration procedure on FlexPendant

Overview

This section describes how to use the FlexPendant when performing a fine calibration of the robot. The method of fitting the calibration equipment to each axis is detailed in the calibration instruction for the axis, see the Product manual for the robot.

Fine calibration procedure

The procedure below details how to perform the fine calibration procedure on the FlexPendant.

Step	Action
1.	 - **Warning!** Do not fine calibrate the robot without special equipment used for axis calibration! It would cause an unsatisfied accuracy in the robot movement.
2.	On the **ABB** menu, tap **Calibration**. All mechanical units connected to the system are shown along with their calibration status.
3.	Tap to select the mechanical unit. A screen is displayed: tap **Calib. Parameter**.  en0400001127

Continues on next page

Step	Action
4.	Tap **Fine Calibration...**. A dialog box is displayed, urging you to use external equipment to performing the actual calibration. Make sure all necessary calibration equipment is fitted, as detailed in the calibration instruction, for the axis to be calibrated. A warning that updating the revolution counters may change programmed robot positions is also displayed: • Tap **Yes** to proceed. • Tap **No** to cancel.
5.	Select the axis to calibrate by ticking the box to the left.
6.	Tap **Calibrate**. A dialog box is displayed, warning that calibration of the selected axes will be changed, which cannot be undone: • Tap **Calibrate** to proceed. • Tap **Cancel** to cancel. Tapping **Calibrate** results in briefly displaying a dialog box, announcing that the calibration process has started. The axis is calibrated and the system returns to the list of available mechanical units.

11.6. Serial Measurement Board memory

Serial Measurement Board, SMB

The Serial Measurement Board, SMB, primarily gathers resolver data from the robot's (or additional axes) motors. This data is used to measure the speed and position of each axis. Each SMB is capable of measuring up to 7 axes. It also stores a number of data pertaining to each robot.

This data is used by the controller and can be transferred between the SMB and the controller. Normally, the data is transferred automatically, but it can also be done manually.

The SMB data is affected when:

- the robot is replaced
- the SMB is replaced
- the controller (or its flash disk) is replaced.
- updating with new calibration data

The following data is stored on the SMB:

- serial number for the mechanical unit
- joint calibration data
- Absolute Accuracy data
- SIS data

Note that if the IRC5 controller is to be connected to a robot with an older SMB, not equipped with data storage capability, the SMB needs to be replaced.

SMB data update

If...	then...
the **flash disk** or the complete **controller** is new or replaced by an unused spare part...	the data stored in the SMB is automatically copied to the controller memory.
the **SMB** is replaced by a new, unused, spare part SMB...	the data stored in the controller memory is automatically copied to the SMB memory.
the **flash disk** or the complete **controller** is replaced by a spare part, previously used in another system...	the data in the controller memory and the SMB memory is different. You must **update the controller memory manually** from the the SMB memory.
the **SMB** is replaced by a spare part SMB, previously used in another system...	the data in the controller memory and the SMB memory is different. You must first **clear the data in the new SMB memory**, and then **update the SMB memory** with the data from the controller memory.
new **calibration data** has been loaded via RobotStudio^{Online} or using the FlexPendant and the system has been restarted...	the data in the controller memory and the SMB memory is different. You must **update the SMB memory manually** from the controller memory. Check that the new calibration values belong to a manipulator with the serial number defined in your system.

Continues on next page

View SMB data status

This section describes how to view the data status in the Serial Measurement Board and the controller.

Step	Action
1.	On the **ABB** menu, tap **Calibration** and select a mechanical unit.
2.	Tap **SMB memory** and then tap **Show status**. The data is displayed with status on the SMB and on the controller.

Update controller data from SMB memory

This section describes how to load data from the Serial Measurement Board **to** the controller.

Step	Action	Info
1.	On the **ABB** menu, tap **Calibration** and select a mechanical unit.	
2.	Tap **SMB memory** and then tap **Update**.	
3.	Tap the button **Cabinet or manipulator has been exchanged**. A warning is displayed. Tap **Yes** to proceed or **No** to cancel.	It is vital that you load calibration data correctly.
4.	The data is loaded. Tap **Yes** to acknowledge and restart the robot system.	The following data is updated: • serial numbers for mechanical units • calibration data • Absolute Accuracy data • SIS data

Update data in SMB memory

This section describes how to update data on the Serial Measurement Board **from** the controller. This is e.g. after calibration data has been loaded to the controller via RobotStudio Online or using the FlexPendant.

If the SMB already contains data, you must first clear the memory, see *Delete SMB data on page 282*.

Step	Action	Info
1.	On the **ABB** menu, tap **Calibration** and select a mechanical unit.	
2.	Tap **SMB memory** and then tap **Update**.	
3.	Tap the button **Serial measurement board has been replaced**. A warning is displayed. Tap **Yes** to proceed or **No** to cancel.	It is vital that you load calibration data correctly.
4.	The data is updated.	

Continues on next page

Delete SMB data

This section describes how to delete the data stored on the SMB memory or the controller memory, when creating spare parts.

Step	Action
1.	On the **ABB** menu, tap **Calibration** and tap to select a mechanical unit.
2.	Tap **SMB memory** and then tap **Advanced**. The following functions are available: • Clear Cabinet Memory • Clear SMB Memory
3.	Tap **Clear Cabinet Memory** if the controller should be replaced and used as a spare part. A list of the SMB data stored in the controller is displayed. Tap **Clear** to delete the memory for the selected robot. Repeat the procedure for all robots in the controller memory.
4.	Tap **Clear SMB Memory** if the SMB should be replaced and used as a spare part. A list of the SMB data stored is displayed. Tap **Clear** to delete the memory for the selected robot. Repeat the procedure for all robots using this SMB board.

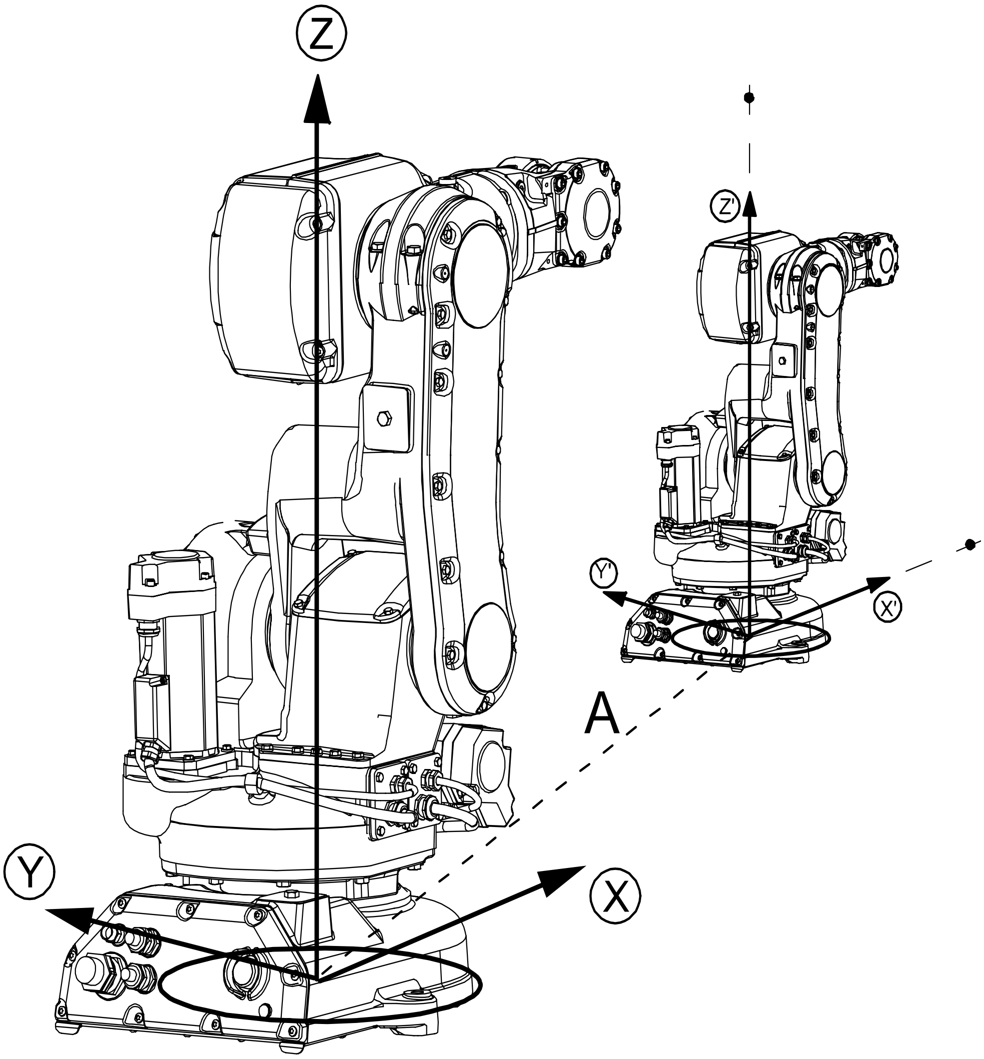
11.7. 4 points XZ calibration

Base Frame calibration

This section describes the 4 points XZ calibration, in the Base Frame calibration options. Other calibration methods may be available in this menu depending on your installed options.

4 points XZ calibration

This section describes how to define the base frame using the 4 points XZ method. This method means that you displace the base frame from the world frame a specified amount in three dimensions and two planes.



xx0400000782

A	Displacement distance between base frame and world frame
X	X-axis in the base frame
Y	Y-axis in the base frame
Z	Z-axis in the base frame
X’	X-axis in the world frame

Continues on next page

Y’	Y-axis in the world frame
Z’	Z-axis in the world frame

Step	Action	Info
1.	On the **ABB** menu, tap **Calibration** and select a mechanical unit. Then tap **Base Frame**.	
2.	Tap **4 points XZ...**.	
3.	Tap **...** to change reference point. A numerical keyboard and boxes for X, Y and Z values are displayed .	
4.	Is a file available with all transformation data? If YES: Tap the **Positions** menu and then tap **Load**. Then select the file containing the values. Load the file. If NO: Proceed to the next step.	
5.	Enter the previously measured displacement in each coordinate box. Tap **OK** when done.	You have now specified the amount of displacement the base frame will be moved from the world frame. Proceed below to specify the direction of the new base frame in relation to the world frame.
6.	Set up a fixed reference position within the working range of the robot, e.g. the tip of a pen secured to the work bench.	
7.	Tap **Point 1** to highlight the line.	
8.	Manually run the robot to the previously fixed reference point.	
9.	Tap **Modify position**. Modified is displayed on the status line.	
10.	Re-orient the robot and again, run it to the reference point but from a different angle.	Repeat these steps until points 1, 2, 3, and 4 have been modified.
11.	Tap **Elongator X** to highlight the line.	
12.	Manually run the robot to a position where the tool center point (TCP) touches an imaginary extension of the X-axis.	The imaginary X-axis is shown in the illustration above.
13.	Tap **Modify position**. Modified is displayed on the status line.	Repeat these steps to modify **Elongator Z**.
14.	To save the entered transformation data to a file, tap the **Positions** menu and then **Save**. Enter the name of the file and then tap **OK**.	
15.	To delete all entered transformation data, tap the **Positions** menu and then **Reset All**.	

12 Changing FlexPendant settings

12.1. Changing brightness and contrast

Appearance options

This section describes the Appearance menu, where you can adjust the screen’s brightness and contrast.

Changing brightness and contrast

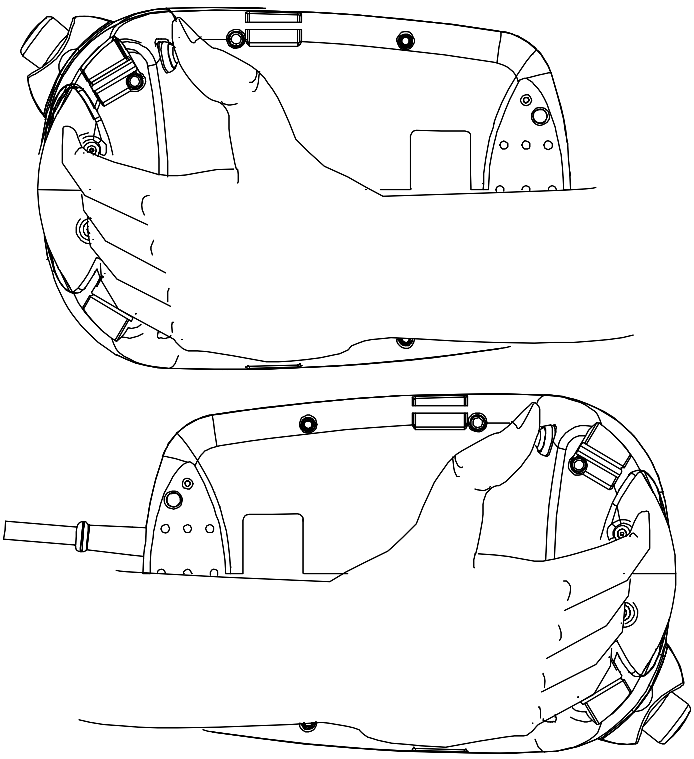
This procedure describes how to change brightness and contrast of the screen.

Step	Action
1.	On the **ABB** menu, tap **Control Panel**.
2.	Tap **Appearance**.
3.	Tap the appropriate **Plus** or **Minus** button to adjust the levels. Tap **Set Default** to return to the default levels. The brightness and contrast changes as you change the levels which gives you an instant view of how the new levels will affect the visibility.
4.	Tap **OK** to use the the new brightness and contrast levels.

12.2. Switching between left and right handheld FlexPendant

Left and right handheld

The FlexPendant is set to left handheld on delivery. This can easily be changed to right handheld and back again whenever required.



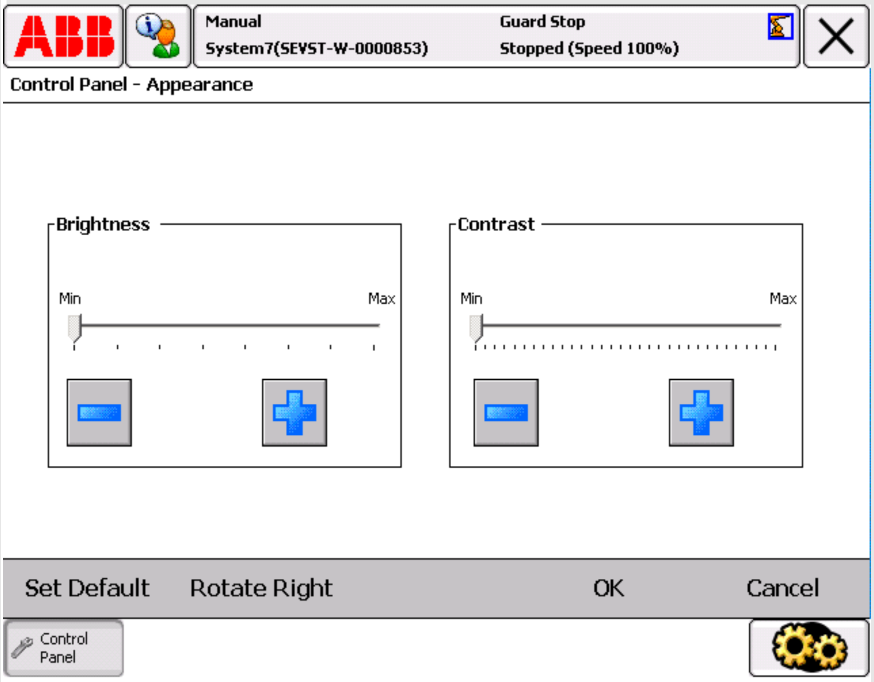
en0400000913

Switching between left and right handheld

This section details how to switch between right and left handheld FlexPendant.

Step	Action
1.	Tap the **ABB** menu, then tap **Control Panel**.
2.	Tap **Appearance**.

Continues on next page

Step	Action
3.	Tap **Rotate right** (or **Rotate left** if the FlexPendant is set for right handheld).  en0400000915
4.	Rotate the FlexPendant and switch to your other arm.

Changes in settings for right handheld FlexPendant

The following settings are effected when changing the FlexPendant to right handheld.

Setting	Effect	Information
Jogging directions	The joystick directions are adjusted automatically.	The illustrations of jogging directions in the jogging menu are adjusted automatically for the present left/right mode.
Hardware buttons and programmable keys	Start, Stop, Forward, and Backward buttons **do not** change place with programmable keys.	See buttons A-G in the illustration *Hardware buttons on page 39*.
Hold-to-run button	No effect	
Enabling device	No effect	

12.3. Changing date and time

Changing date and time

This procedure details how to set the controller clock.

Step	Action
1.	On the **ABB** menu, tap **Control Panel**.
2.	Tap **Date and Time**. The current date and time is displayed.
3.	Tap the appropriate **Plus** or **Minus** button to change the date or time.
4.	Tap **OK** to use the time and date settings.

NOTE!

The date and time is always displayed according to ISO standard, that is, year-month-day and hour:minute, the time using 24-hour format.

12.4. Configuring Most Common I/O

Most Common I/O

Most Common I/O is used in the Program Editor to display a list of the most commonly used I/O signals in the robot system. Since there can be many signals, it may be very helpful to be able to make this selection.

The sorting in the list can be rearranged manually. By default, the signals are sorted in the order that they are created.

Most Common I/O can also be configured using system parameters in the topic Man-machine Communication. However, sorting the list can only be done by using the function under the Control Panel. See section *Configuring system parameters on page 270*.

Configuring Most Common I/O

This section describes how to configure the list Most Common I/O.

Step	Action
1.	On the **ABB** menu, tap **Control Panel**.
2.	Tap **I/O**. A list of all I/O signals defined in the system is listed with checkboxes.
3.	Tap the names of the signals to select for the Most Common I/O list. Tap **All** or **None** to select all or no signals. Tap **Name** or **Type** to sort by name or signal type.
4.	Tap **Preview** to see the list of selected signals and adjust the sort order. Tap to select a signal and then tap the arrows to move the signal up or down in the list, rearranging the sort order. Tap **APPLY** to save the sort order. Tap **Edit** to return to the list of all signals.
5.	Tap **APPLY** to save the settings.

12.5. Changing language

Languages

This procedure details how to change between the currently installed languages. The FlexPendant supports up to three languages at any one time.

After selecting a specific language, all buttons, menus and dialogs are displayed using this language. RAPID instructions, variables, system parameters, and I/O signals are not affected.

Changing language

This section describes how to change language on the FlexPendant.

Step	Action
1.	On the **ABB** menu, tap **Control Panel**.
2.	Tap **Language**. A list of all installed languages is displayed.
3.	Tap the language that you want to change to.
4.	Tap **OK**. A dialog box is displayed. Tap **Yes** to proceed and restart the FlexPendant. The current language is replaced by the selected. All buttons, menus and dialogs are displayed using the new language.

12.6. Editing programmable keys

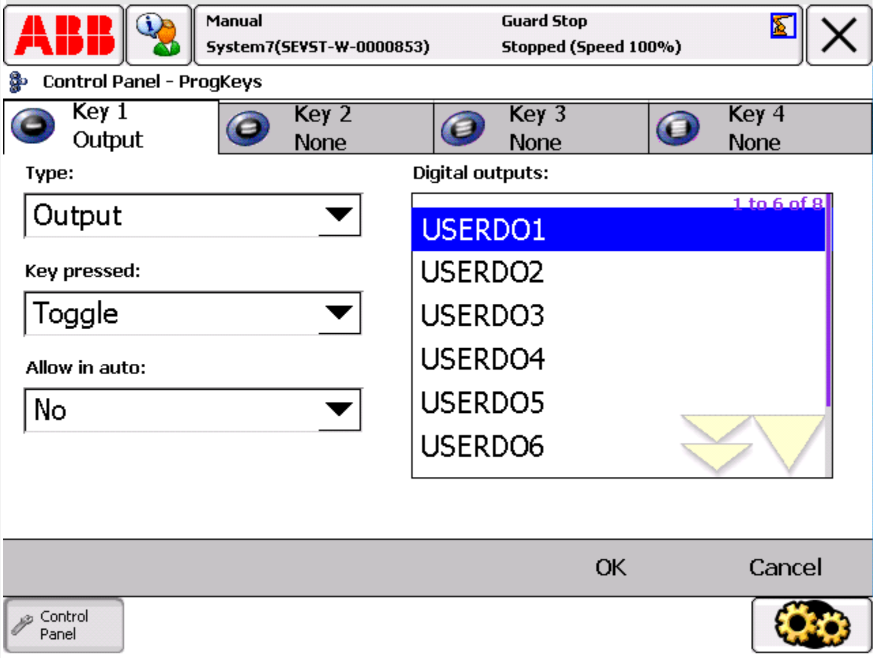
Overview

Programmable keys are four hardware buttons on the FlexPendant that may be used for dedicated, specific functions set by the user. See *Hardware buttons on page 39*.

The keys can be programmed to simplify programming or testing of programs. They can also be used to activate menues on the FlexPendant.

Editing programmable keys

This section describes how to edit the programmable keys.

Step	Action
1.	On the **ABB** menu, tap **Control Panel**.
2.	Tap **ProgKeys**.  en0400001154
3.	Select key to edit, Key 1-4 in the upper selection list.
4.	Tap the **Type** menu to select type of action: • None • Input • Output • System
5.	If Type **Input** is selected. • Tap to select one of the digital inputs from the list. • Tap the **Allow in auto** menu to select if the function is also allowed in automatic operating mode.

Continues on next page

Step	Action
6.	If Type **Output** is selected. • Tap to select one of the digital outputs from the list • Tap the **Key pressed** menu to define how the signal should behave when the key is pressed. • Tap the Allow in auto menu to select if the function is also allowed in automatic operating mode Key pressed functions: • Toggle - switches signal value from 0 to 1 or vice versa • Set to 1 - sets the signal to 1 • Set to 0 - sets the signal to 0 • Press/Release - sets signal value to 1 while key is pressed (note that an inverted signal will be set to 0) • Pulse - the signal value pulses once
7.	If Type **System** is selected. • Tap the **Key pressed** menu to select Move PP to main • Tap the **Allow in auto** menu to select if the function is also allowed in automatic operating mode
8.	Edit the other keys as described in steps 3 to 7 above.
9.	Tap **OK** to save the settings.

12.7. Editing supervision settings

Overview

The motion supervision monitors the robot, and stops the robot if a mechanical drag larger than 300 is detected in the motors. The mechanical drag is a value without any specific unit, with a max value of 300 and a min value of 0.

The motion supervision must be set for each task separately.

Description of functions in RobotWare base:

- Path Supervision in automatic mode. This is used to prevent mechanical damage due to running into an obstacle during program execution with robot movement.
- Execution Settings. *Non motion execution* ON is used to run a program without robot motion.

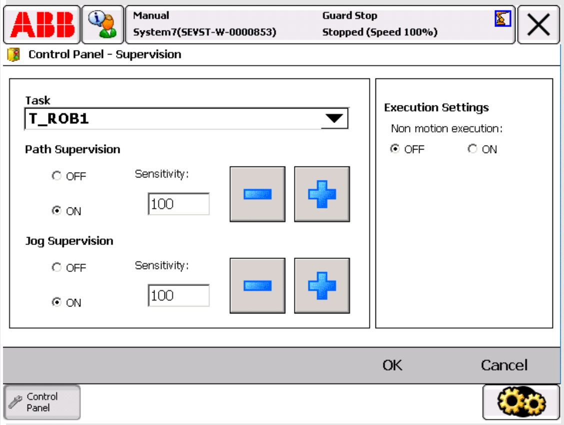
A RobotWare system with the option *Collision Detection* has additional functionality:

- Path Supervision in manual mode and manual full speed mode.
- Jog Supervision. This is used to prevent mechanical damage to the robot during jogging.
- Changing the supervision from a RAPID program.

For more information on *Collision Detection*, see *Application manual - Motion coordination and supervision*.

Editing motion supervision and execution settings

This section describes how to edit the motion supervision and execution settings.

Step	Action	Info
1.	On the **ABB** menu tap **Control Panel** and then **Supervision**.  en0400000970	
2.	Tap the **Task** menu to select a task for the motion settings.	The settings only apply for one task. If you have more than one task, you need to set the desired values for each task separately.

Continues on next page

Step	Action	Info
3.	In the Path Supervision settings, tap **ON** or **OFF** to apply or remove path supervision. Tap the **+** or **-** buttons to set the sensitivity. Note: unless you have the option *Collision Detection* installed, this only affects the robot in auto mode.	TIP! The sensitivity may be adjusted from 0 to 300. Do not set the motion sensitivity lower than 80, or the robot will stop due to internal drag.
4.	In the Jog Supervision settings, tap **ON** or **OFF** to apply or remove jog supervision. Tap the **+** or **-** buttons to set the sensitivity. Note: this only affects the robot if you have the option *Collision Detection* installed.	TIP! The sensitivity may be adjusted from 0 to 300. Do not set the motion sensitivity lower than 80, or the robot will stop due to internal drag.
5.	Under the Execution Settings, Non motion exectution may be turned **ON** or **OFF**.	Non motion execution is described in section *Non motion execution* below.

Non motion execution

Non motion execution is a function that enables running a RAPID program without robot motion. All other functions work normally; current cycle times, I/O, TCP speed calculation etc.

This function is mainly used for program debugging, cycle time evaluation, and possibility to measure e.g. glue or paint consumption during a cycle.

Non motion execution is set from the FlexPendant. The function can only be set if the system is in Motors Off state.

When non motion execution is turned on, it can be executed in:

- Manual mode
- Manual mode full speed
- Auto mode

Cycle times will be simulated according to the selected mode.

WARNING!

Non-motion execution is reset after a reboot. Do not restart the program without checking status of the Non-motion execution. Starting the program incorrectly may cause serious injury or death, or damage the robot or other equipment.

12.8. Configuring view settings

FlexPendant system configurations

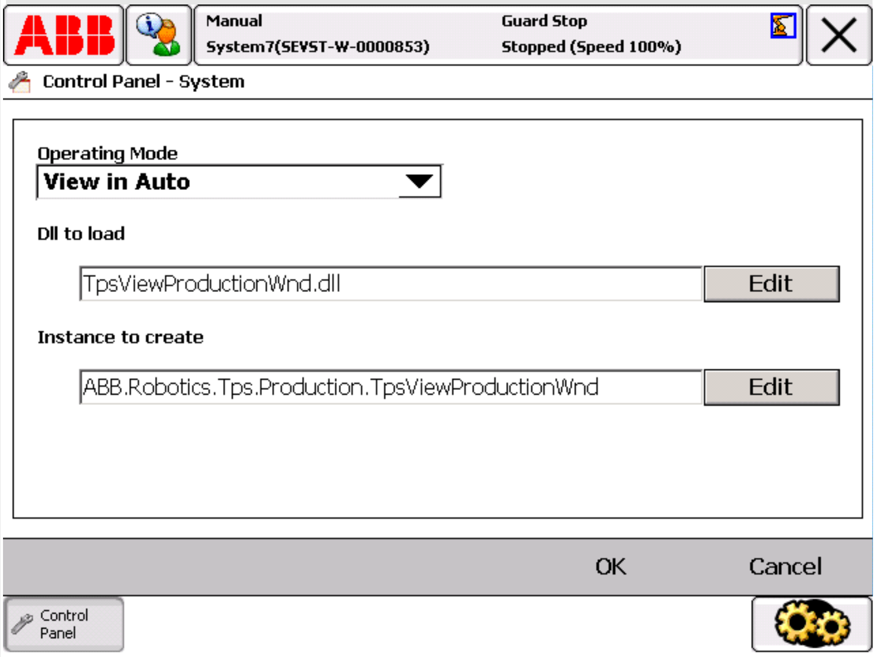
The FlexPendant system configuration is used to control views for operating modes and the user authorization system.

View on Operating Mode change

This section describes how to configure the FlexPendant system for which views to show when changing operating mode.

This is used, for example, when a view other than the Production Window is desired when changing to automatic operating mode.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>On the ABB menu, tap Control Panel and then tap System.</td></tr>
<tr><td>2.</td><td>Tap View on Operating Mode change.

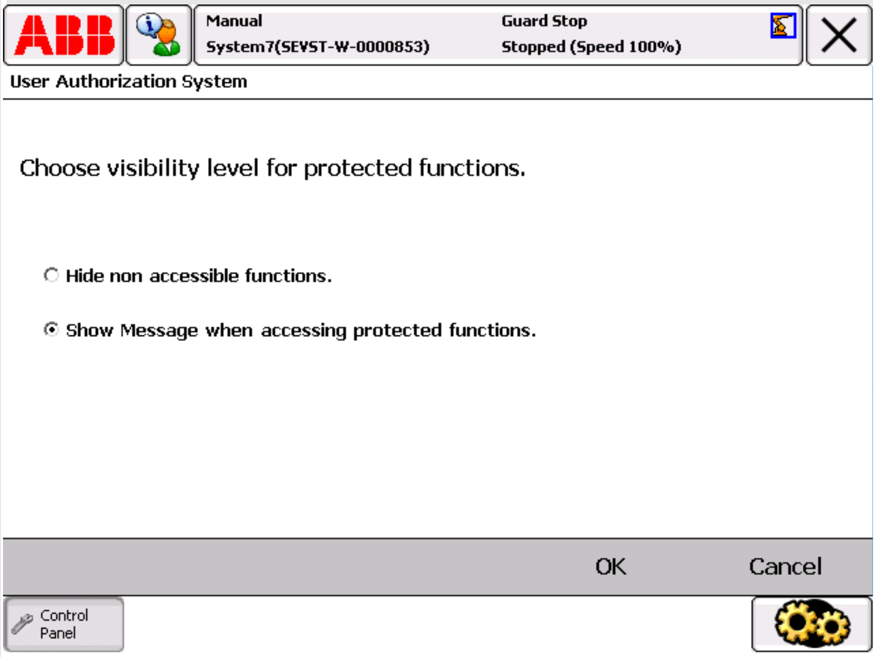


en0400001152</td></tr>
<tr><td>3.</td><td>Tap the Operating Mode menu to select:
• View in Auto
• View in Manual
• View in Manual Full Speed
All three modes can be defined.</td></tr>
<tr><td>4.</td><td>Tap Edit to define the names for the dll and the instance to create.
The dll can contain a number of instances.</td></tr>
<tr><td>5.</td><td>Tap OK.</td></tr>
</table>

Continues on next page

User Authorization System protected function visibility

This section describes how to change the visibility of protected functions for the user authorization system, UAS. The protected functions can be hidden or displayed but not accessible. All other administration of the user authorization system is done using RobotStudio$^{\text{Online}}$.

Step	Action
1.	On the **ABB** menu, tap **Control Panel** and then tap **System**.
2.	Tap **User Authorization System**.  en0400001153
3.	Tap to select the level of visibility for User Authorization protected functions: • Hide non accessible functions • Show message when accessing protected functions.
4.	Tap **OK**.

12.9. Changing background image

Background images

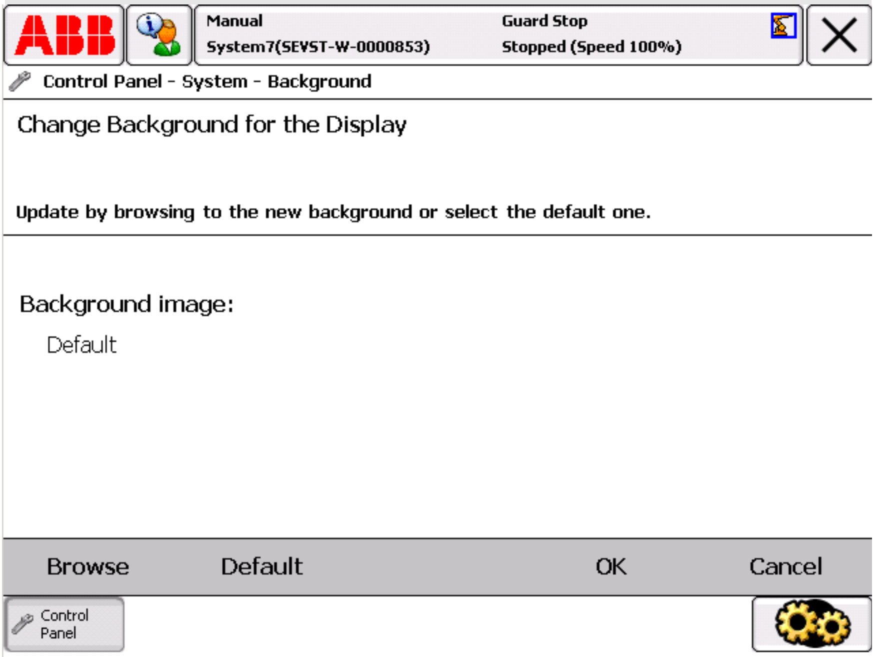
The background image on the FlexPendant can be changed. Any image file on the controller hard disk can be used, a photo as well as an illustration.

For best result, use an image following these recommendations:

- 640 by 390 pixels (width, height)
- Format gif

Changing background image

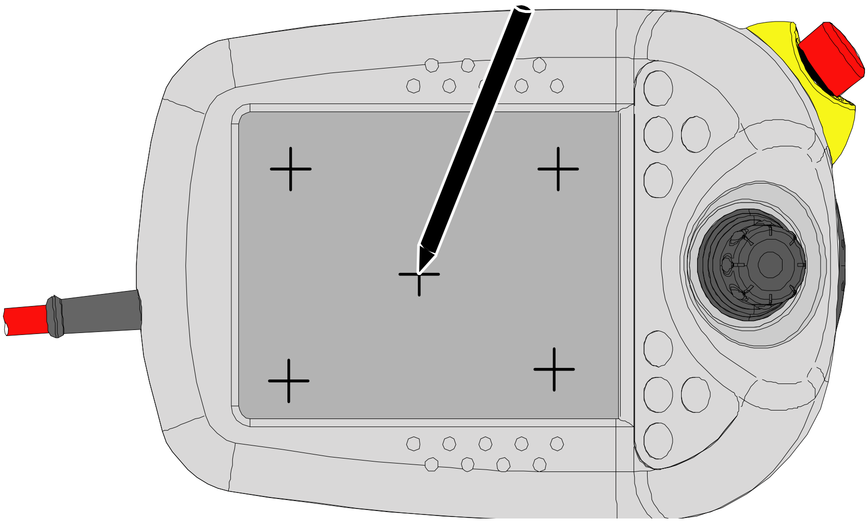
This procedure describes how to change background image on FlexPendant.

Step	Action
1.	On the **ABB** menu, tap **Control panel**.
2.	Tap **System** and then **Background**.  en0500001547
3.	Tap **Browse** to locate another picture on the controller hard disk.
4.	Tap **Default** to restore the original background image.
5.	Tap **OK** to apply the new image or **Cancel** to leave the background unchanged.

12.10. Calibrating the touch screen

Recalibration

This section describes how to recalibrate the touch screen. The touch screen is calibrated on delivery and normally never needs to be recalibrated.



en0400000974

Step	Action	Info
1.	On the **ABB** menu, tap **Control Panel**.	
2.	Tap **Touch Screen**.	
3.	Tap **Recalibrate**. The screen will go blank for a few seconds. A series of crosses will appear on the screen, one at a time.	
4.	Tap the center of each cross with a pointed object.	 Caution Do not use a sharp object which may damage the surface of the screen.
5.	The recalibration is complete.	

13 Descriptions of terms and concepts

13.1. About the Descriptions of terms and concepts chapter

Overview

This chapter contains descriptions of many of the concepts and words used in this manual.

Note that there may also be additional information in any of the chapters dealing with the feature at hand.

13.2. What is the robot system?

Description

The concept *robot system* comprises the manipulator(s), control module, drive module, and all equipment controlled by the controller (tool, sensors, etc.). It includes all hardware as well as software required to operate the robot. Application specific hardware and software, such as spot welding equipment, is not included in the term.

13.3. What are robots, manipulators and positioners?

Manipulator

Manipulator is a generic term for mechanical units used to move objects, tools, etc. The term *manipulator* includes robot as well as positioner.

Robot

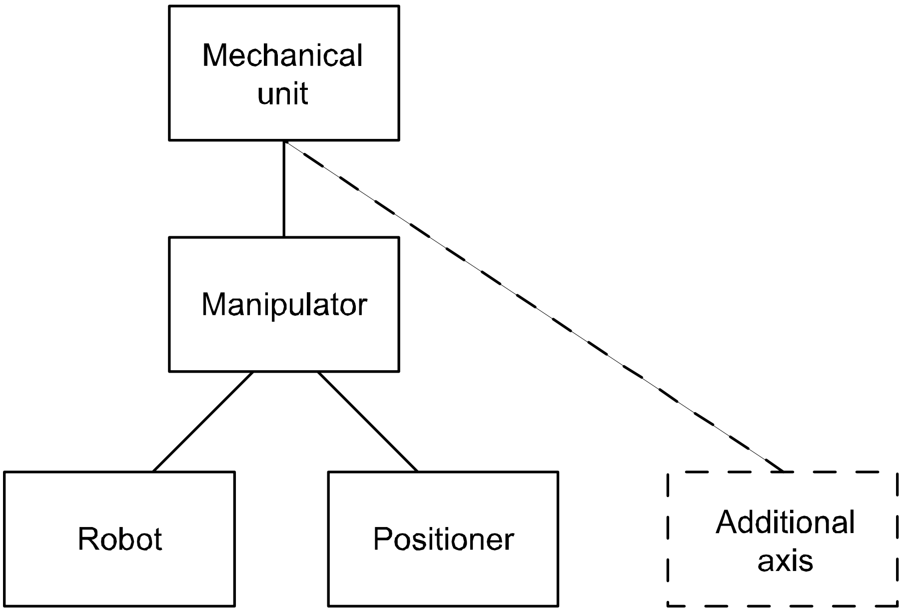
A *robot* is a mechanical unit with TCP. The term robot does not include the controller.

Positioner

A *positioner* is a mechanical unit used to move a work object. It may have one or several axes, normally no more than 3 axes. A positioner normally does not have a TCP.

Illustration

The illustration depicts the relation between the concepts robot, manipulator, positioner, mechanical unit, and other units, e.g. external axes.



en0400000940

Mechanical unit

A *mechanical unit* can be jogged, it can either be a robot, a single additional axis or a set of external axes, for instance a two axis positioner.

13.4. What is a tool?

Tool

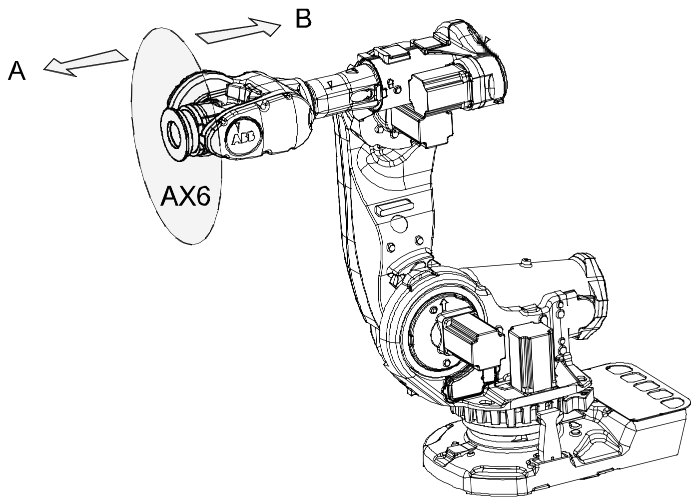
A tool is an object that can be mounted directly or indirectly on the robot turning disk or fitted in a fixed position within the robot working range.

A fixture (jig) is not a tool.

All tools must be defined with a TCP (Tool Center Point).

Each tool that may be used by the robot must be measured and its data stored in order to achieve accurate positioning of the tool center point.

Illustration



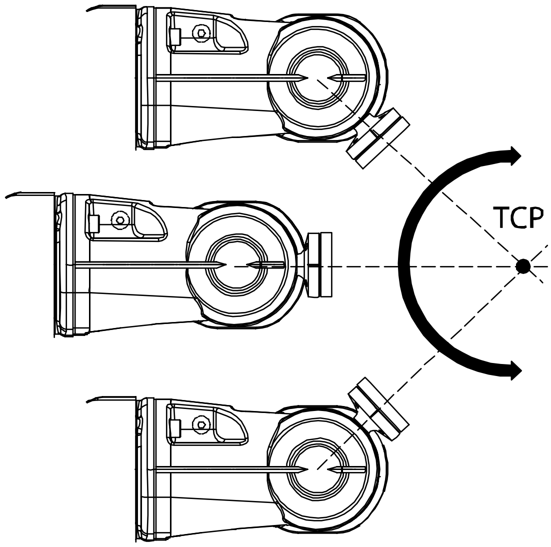
en0400000803

A	Tool side
B	Robot side

13.5. What is the tool center point?

Illustration

The illustration shows how the tool center point (TCP) is the point around which the orientation of the tool/manipulator wrist is being defined.



xx0300000604

Description

The tool center point (TCP) is the point in relation to which all robot positioning is defined. Usually the TCP is defined as relative to a position on the manipulator turning disk.

The TCP will be jogged or moved to the programmed target position. The tool center point also constitutes the origin of the tool coordinate system.

The robot system can handle a number of TCP definitions, but only one may be active at any one time.

There are two basic types of TCPs: moveable or stationary.

Moving TCP

The vast majority of all applications deal with moving TCP, i.e. a TCP that moves in space along with the manipulator.

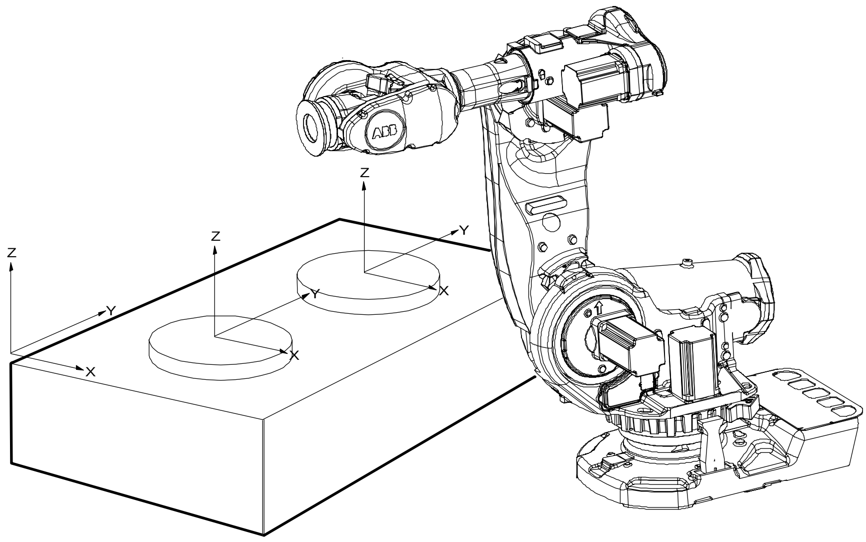
A typical moving TCP may be defined in relation to, e.g. the tip of a arc welding gun, the center of a spot welding gun or the end of a grading tool.

Stationary TCP

In some applications a stationary TCP is used, e.g. when a stationary spot welding gun is used. In such cases the TCP may be defined in relation to the stationary equipment instead of the moving manipulator.

13.6. What is a work object?

Illustration



en0400000819

Description

A work object is a coordinate system with specific properties attached to it. It is mainly used to simplify programing when editing programs due to displacements of specific tasks, objects processes etc.

The work object coordinate system must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).

Work objects are often created to simplify jogging along the object's surfaces. There might be several different work objects created so you must choose which one to use for jogging.

Payloads are important when working with grippers. In order to position and manipulate an object as accurate as possible its weight must be accounted for. You must choose which one to use for jogging.

13.7. What is a coordinate system?

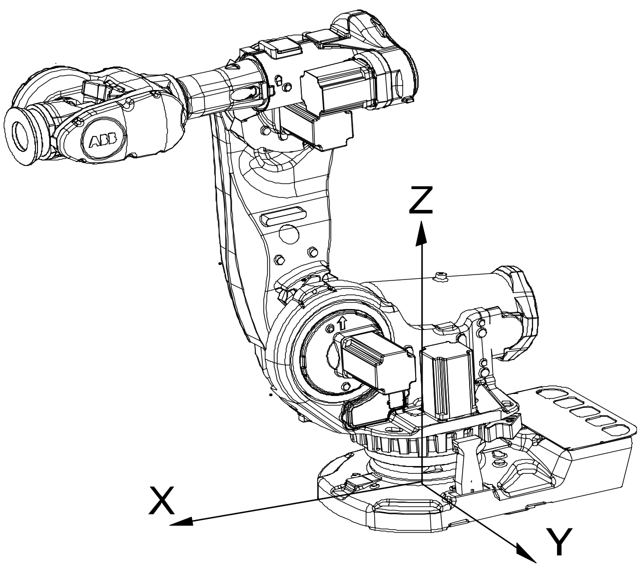
Overview

A coordinate system defines a plane or space by axes from a fixed point called the origin. Robot targets and positions are located by measurements along the axes of coordinate systems.

A robot uses several coordinate systems, each suitable for specific types of jogging or programming.

- The *base coordinate system* is located att the base of the root.It is the easiest one for just moving the robot from one position to another. See *The base coordinate system on page 305*for more information.
- The *work object coordinate system* is related to the work piece and is often the best one for programming the robot. See *The work object coordinate system on page 307*for more information.
- The *tool coordinate system* defines the position of the tool the robot uses when reaching the programmed targets. See *The tool coordinate system on page 309*for more information.
- The *world coordinate system*that defines the robot cell, all other coordiante systems are realted to the the world coordinate system, either directly or indirectly. it is useful for jogging, genral movements and for handling stations and cells with several robots or robots moved by external axes. See *The world coordinate system on page 306*for more information.
- The u*ser coordinate system* is useful for representing equipment that holds other coordinate systems, like work objects. See *The user coordinate system on page 310* for more information.

The base coordinate system



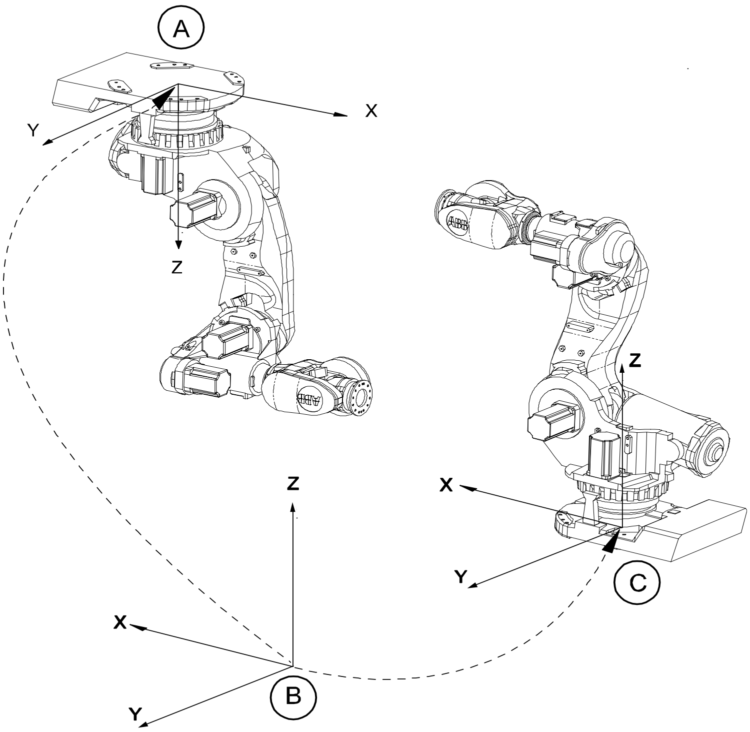
xx0300000495

Continues on next page

The base coordinate system has its zero point in the base of the robot, which makes movements predictable for fixed mounted robots. It is therefore useful for jogging a robot from one position to another. For programming a robot, other coordinate systems, like the work object coordinate system are often better choices. See *The work object coordinate system on page 307*for more information.

When you are standing in front of the robot and jog in the base coordinate system, in a normally configured robot system, pulling the joystick towards you will move the robot along the X axis, while moving the joystick to the sides will move the robot along the Y axis. Twisting the joystick will move the robot along the Z axis.

The world coordinate system



en0300000496

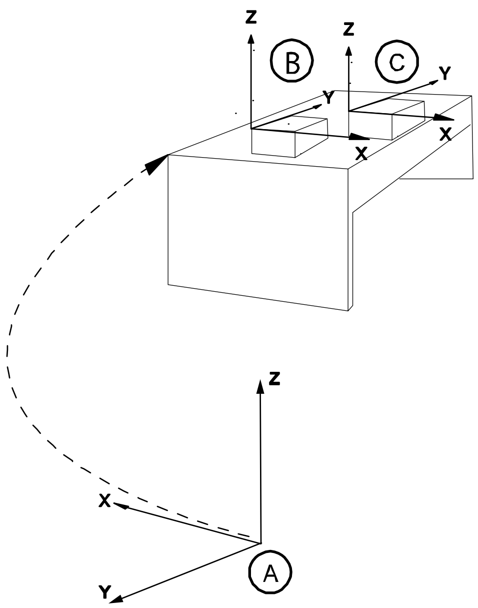
A	Base coordinate system for robot 1
B	World coordinate
C	Base coordinate system for robot 2

The world coordinate system has its zero point on a fixed position in the cell or station. This makes it useful for handling several robots or robots moved by external axes.

By default the world coordinate system coincidences with the base coordinate system.

Continues on next page

The work object coordinate system



en0300000498

A	World coordinate system
B	Work Object coordinate system 1
C	Work Object coordinate system 2

The work object coordinate system corresponds to the work piece: It defines the placement of the work piece in relation to the world coordinate system (or any other coordinate system).

The work object coordinate system must be defined in two frames, the user frame (related to the world frame) and the object frame (related to the user frame).

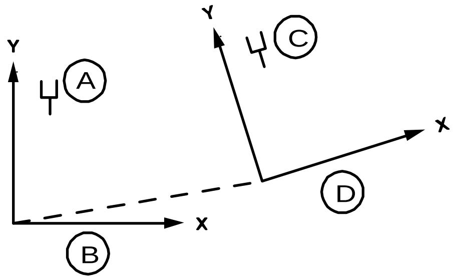
A robot can have several work object coordinate systems, either for representing different work pieces or several copies of the same work piece at different locations.

It is in work object coordinate systems you create targets and paths when programming the robot. This gives a lot of advantages:

- When repositioning the work piece in the station you just change the position of the work object coordinate system and all paths are updated at once.
- Enables work on work pieces moved by external axes or conveyor tracks, since the entire work object with its paths can be moved.

Continues on next page

The displacement coordinate system



en0400001227

A	Original position
B	Object coordinate system
C	New position
D	Displacement coordinate system

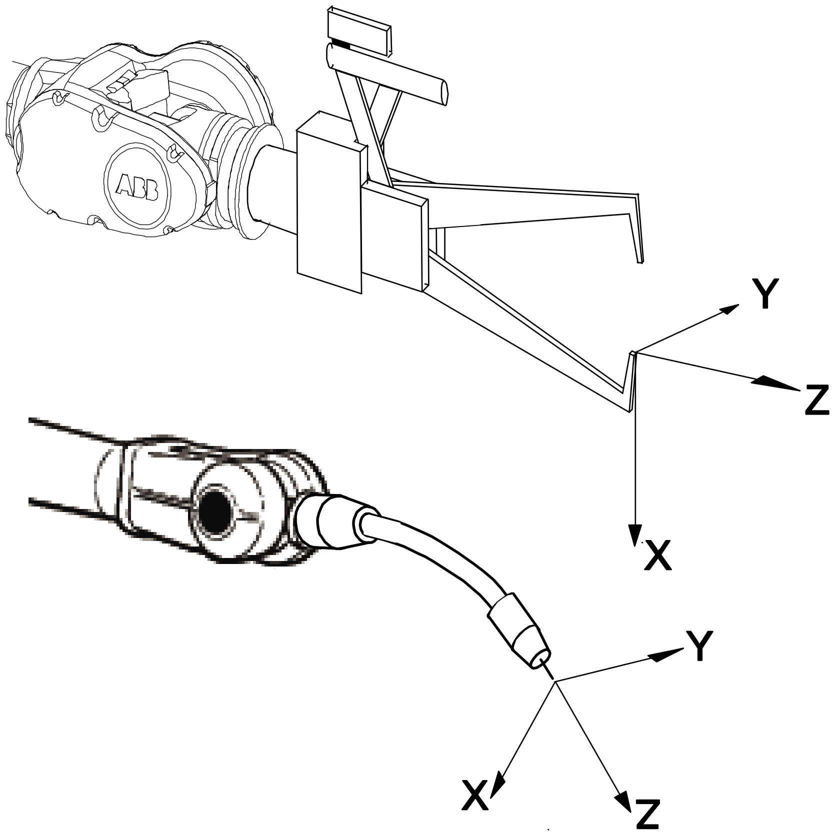
Sometimes, the same path is to be performed at several places on the same objec, or on several work pieces located next to each opther. To avoid having to reprogram all positions each time a displacement coordinate system can be defined.

This coordinate system can also be used in conjunction with searches, to compensate for differences in the positions of the individual parts.

The displacement coordinate system is defined based on the object coordinate system.

Continues on next page

The tool coordinate system



xx0300000506

The tool coordinate system has its zero position at the center point of the tool. It thereby defines the position and orientation of the tool. The tool coordinate system is often abbreviated to TCP (Tool Center Point) or TCPF (Tool Center Point Frame).

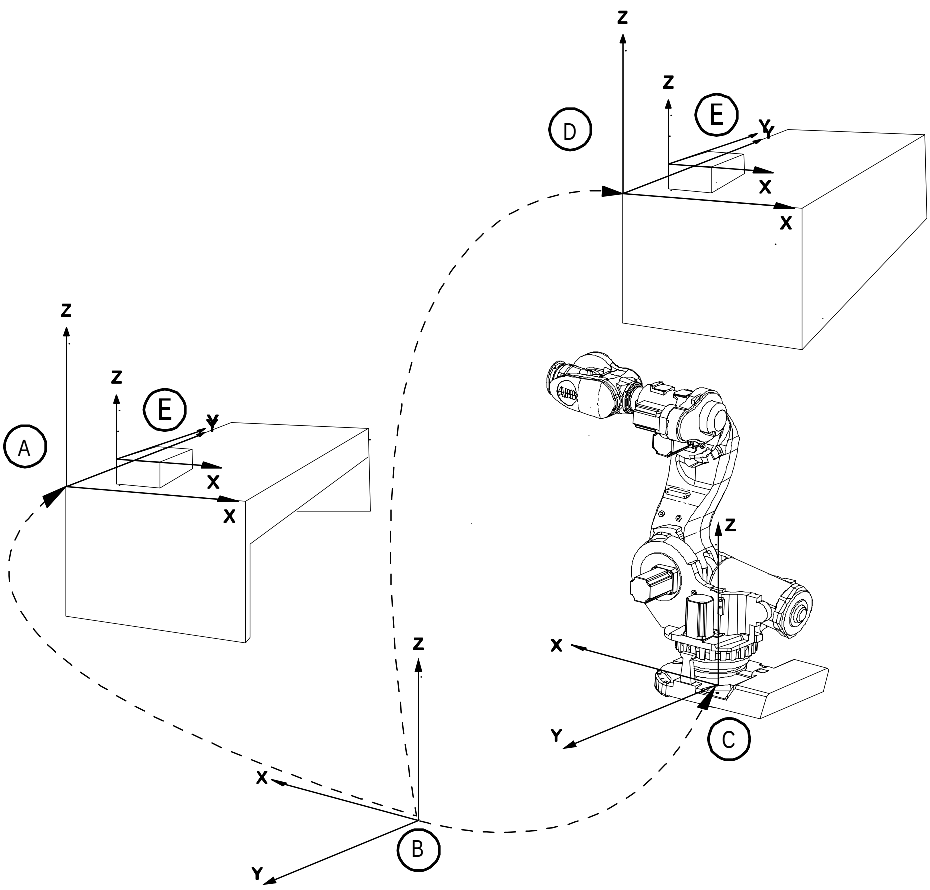
It is the TCP the robot moves to the programmed positions, when executing programs. This means that if you change the tool (and the tool coordinate system) the robot's movements will be changed so that the new TCP will reach the target.

All robots have a predefined tool coordinate system, called *tool0*, located at the wrist of the robot. One or many new tool coordinate systems can then defined as offsets from *tool0*.

When jogging a robot the tool coordinate system is useful when you don't want to change the orientation of the tool during the movement, for instance moving a saw blade without bending it.

Continues on next page

The user coordinate system



en0300000497

A	User coordinate system
B	World coordinate system
C	Work object coordinate system
D	Moved user coordinate system
E	Work object coordinate system, moved with user coordinate system

The user coordinate system can be used for representing equipment like fixtures, workbenches. This gives an extra level in the chain of related coordinate systems, which might be useful for handling equipment that hold work objects or other coordinate systems.

13.8. What is a RAPID application?

Purpose

A RAPID application contains a sequence of instructions that controls the robot so that it may perform the operations it is intended for.

Contents of the RAPID application

An application is written using a particular vocabulary and syntax called *RAPID programming language*.

The programming language contains instructions in English enabling the robot to move, setting outputs and reading inputs. It also contains instructions for making decisions, to repeat other instructions, to structure the program, to communicate with the system operator and more.

Structure of the RAPID application

The structure of a RAPID application is shown in section *The structure of a RAPID application on page 134*.

How is an application stored?

An application you work with or run must be loaded in the controller's program memory. This procedure is called to *Load* the application.

You *Save* applications on the controller's hard disk or other disk memory to keep them safe when you want to work on another application.

See also *What is "the memory"? on page 248*.

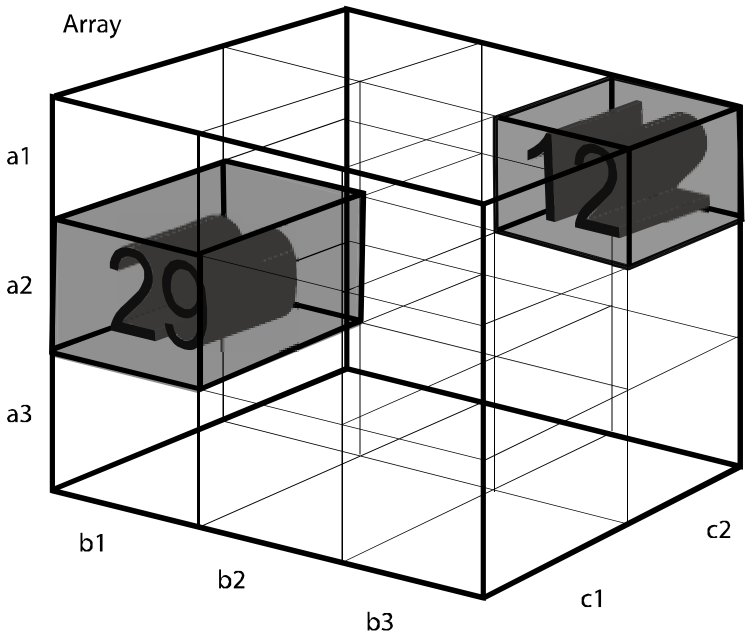
13.9. What is a data array?

Overview

A data array is a special type of variable: a regular variable may contain one data value, but an array may contain several.

It may be construed as a table, which may have one or more dimensions. This table may be populated with data (e.g. numerical values or character strings) to be used during programming or operation of the robot system.

An example of a three dimensional array is shown below:



en0400001006

This array, called "Array" is defined by its three dimensions a, b and c. Dimension a has three rows, b has three rows (columns) and c has two rows. The array and its contents may be written as `Array {a, b, c}`.

Example 1: `Array {2, 1, 1}=29`

Example 2: `Array {1, 3, 2}=12`